Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek F

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume F

Robotika
Robotics

Uredili / Edited by
Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams

http://is.ijs.si

8. oktober 2014 / October 8th, 2014
Ljubljana, Slovenia

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek F

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume F

Robotika

Robotics

Uredili / Edited by

Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams

http://is.ijs.si

8. oktober 2014 / October 8th, 2014
Ljubljana, Slovenia

Uredniki:

Jadran Lenarčič
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Aleš Ude
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Bojan Nemec
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Andrej Gams
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič, Mitja Lasič

Ljubljana, oktober 2014

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

007.52(082)(0.034.2)

MEDNARODNA multikonferenca Informacijska družba (17 ; 2014 ; Ljubljana)
Robotika [Elektronski vir] : zbornik 17. mednarodne multikonference Informacijska družba - IS 2014, 8. oktober 2014, [Ljubljana, Slovenia] : zvezek F = Robotics : proceedings of the 17th International Multiconference Information Society - IS 2014, October 8th, 2014, Ljubljana, Slovenia : volume F / uredili, edited by Jadran Lenarčič ... [et al.]. - El. knjiga. - Ljubljana : Institut Jožef Stefan, 2014

Način dostopa (URL): http://library.ijs.si/Stacks/Proceedings/InformationSociety

ISBN 978-961-264-076-7 (pdf)
1. Gl. stv. nasl. 2. Vzp. stv. nasl. 3. Dodat. nasl. 4. Lenarčič, Jadran
275926784

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2014

Multikonferenca Informacijska družba (http://is.ijs.si) s sedemnajsto zaporedno prireditvijo postaja tradicionalna kvalitetna srednjeevropska konferenca na področju informacijske družbe, računalništva in informatike. Informacijska družba, znanje in umetna inteligenca se razvijajo čedalje hitreje. Čedalje več pokazateljev kaže, da prehajamo v naslednje civilizacijsko obdobje. Npr. v nekaterih državah je dovoljena samostojna vožnja inteligentnih avtomobilov, na trgu pa je moč dobiti kar nekaj pogosto prodajanih tipov avtomobilov z avtonomnimi funkcijami kot »lane assist«. Hkrati pa so konflikti sodobne družbe čedalje bolj nerazumljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc in delavnic. Predstavljenih bo okoli 200 referatov, prireditev bodo spremljale okrogle mize, razprave ter posebni dogodki kot svečana podelitev nagrad. Referati so objavljeni v zbornikih multikonference, izbrani prispevki bodo izšli tudi v posebnih številkah dveh znanstvenih revij, od katerih je ena Informatica, ki se ponaša s 37-letno tradicijo odlične evropske znanstvene revije.

Multikonferenco Informacijska družba 2014 sestavljajo naslednje samostojne konference:

- Inteligentni sistemi
- Izkopavanje znanja in podatkovna skladišča
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Soočanje z demografskimi izzivi
- Vzgoja in izobraževanje v informacijski družbi
- Kognitivna znanost
- Robotika
- Jezikovne tehnologije
- Interakcija človek-računalnik v informacijski družbi
- Prva študentska konferenca s področja računalništva
- Okolijska ergonomija in fiziologija
- Delavnica Chiron.

Soorganizatorji in podporniki konference so različne raziskovalne in pedagoške institucije in združenja, med njimi tudi ACM Slovenija, SLAIS in IAS. V imenu organizatorjev konference se želimo posebej zahvaliti udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V 2014 bomo drugič podelili nagrado za življenjske dosežke v čast Donalda Michija in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe je prejel prof. dr. Janez Grad. Priznanje za dosežek leta je pripadlo dr. Janezu Demšarju. V letu 2014 četrtič podeljujemo nagrado »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobila nerodna izvedba piškotkov, jagodo pa Google Street view, ker je končno posnel Slovenijo. Čestitke nagrajencem!

Niko Zimic, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2014

The Information Society Multiconference (http://is.ijs.si) has become one of the traditional leading conferences in Central Europe devoted to information society. In its 17th year, we deliver a broad range of topics in the open academic environment fostering new ideas which makes our event unique among similar conferences, promoting key visions in interactive, innovative ways. As knowledge progresses even faster, it seems that we are indeed approaching a new civilization era. For example, several countries allow autonomous card driving, and several car models enable autonomous functions such as "lane assist". At the same time, however, it is hard to understand growing conflicts in the human civilization.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers, presented in twelve independent events. The papers are published in the Web conference proceedings, and a selection of them in special issues of two journals. One of them is Informatica with its 37 years of tradition in excellent research publications.

The Information Society 2014 Multiconference consists of the following conferences and workshops:

- Intelligent Systems
- Cognitive Science
- Data Mining and Data Warehouses
- Collaboration, Software and Services in Information Society
- Demographic Challenges
- Robotics
- Language Technologies
- Human-Computer Interaction in Information Society
- Education in Information Society
- 1st Student Computer Science Research Conference
- Environmental Ergonomics and Psysiology
- Chiron Workshop.

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, SLAIS and IAS.

In 2014, the award for life-long outstanding contributions was delivered in memory of Donald Michie and Alan Turing for a second consecutive year. The Programme and Organizing Committees decided to award the Prof. Dr. Janez Grad with the Michie-Turing Award. In addition, a reward for current achievements was pronounced to Prof. Dr. Janez Demšar. The information strawberry is pronounced to Google street view for incorporating Slovenia, while the information lemon goes to cookies for awkward introduction. Congratulations!

On behalf of the conference organizers we would like to thank all participants for their valuable contribution and their interest in this event, and particularly the reviewers for their thorough reviews.

Niko Zimic, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, UK
Olga S. Fomichova, Russia
Vladimir A. Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Izrael
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Finland
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki-Špetič
Mitja Lasič
Robert Blatnik
Mario Konecki
Vedrana Vidulin

Programme Committee

Nikolaj Zimic, chair
Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah
Igor Mekjavić
Tadej Debevec

KAZALO / TABLE OF CONTENTS

Robotika / Robotics ***1***
PREDGOVOR / FOREWORD 3
Active Stereo 3-D Point Reconstruction / Bevec Robert 5
Modulation of Combined Discrete-Periodic Dynamic Movement Primitives with Iterative Learning Control and Repetitive Control / Gams Andrej 9
Mechanical and Control Hardware Design of Humanoid Robot Marko / Borovac Branislav, Raković Mirko, Savić Srđan, Nikolić Milutin, Batinica Aleksandar 13
Motion Primitives for Realization of Robust Humanoid Walk in Unstructured Envioronment / Raković Mirko, Borovac Branislav, Nikolić Milutin, Savić Srđan 17
Force Based Surface Following Based on Iterative Learning / Nemec Bojan, Petrič Tadej, Ude Aleš 21
Configuration Based Complinance Control of Industrial Humanoids / Petrović Petar B., Lukić Nikola, Danilov Ivan 25
On Developing Anthropomimetic Robot-Arm / Rodić Aleksandar, Miloradović Branko, Urukalo Đorđe 29
Taks-Specific Dynamics for Compliant Robot Behavior And Accurate Trajectory Tracking / Deniša Miha, Gams Andrej, Gams Andrej 33

Indeks avtorjev / Author index ***37***

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek F

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume F

Robotika

Robotics

Uredili / Edited by

Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams

http://is.ijs.si

8. oktober 2014 / October 8th, 2014
Ljubljana, Slovenia

PREDGOVOR

Pod okriljem multikonference »Informacijska družba« po letu premora že petič v zadnjih šestih letih zapored organiziramo tudi konferenco Robotika, s katero nadaljujemo tradicijo raziskovalne robotike v Sloveniji.

Čeprav robotiko mnogi še zmeraj dojemajo kot znanstveno fantastiko, pa so raziskovalna, industrijska in pa tudi uporabniška robotika že nekaj časa nekaj povsem realnega in oprijemljivega. Brez robotskih manipulatorjev si ne znamo več predstavljati sodobnih industrijskih procesov. Ne presenečajo niti kirurški roboti ali servisni mobilni roboti, ki bodo kmalu čistili naša stanovanja. Domišljija in pa želje ljudi ne poznajo mej, zato je raziskovalna robotika že zelo blizu preboja, ki bo omogočil uporabo velikih večnamenskih robotskih hišnih pomočnikov. Pri razvoju tako kompleksnih in avtonomnih sistemov, kar nekateri ocenjujejo, da je težje kot raketna znanost, je pomembna izmenjava idej in mnenj, kar je tudi namen konference Robotika.

V zborniku so zbrani prispevki raziskovalcev Odseka za avtomatiko, biokibernetiko in robotiko na Inštitutu Jožef Stefan, veseli pa smo, da imamo letos prispevke s treh raziskovalnih organizacij v Srbiji. Upamo, da bo izmenjava idej in raziskovalnih rezultatov vodila v nadaljnje skupne podvige, ki bodo še naprej pomagali soustvarjati trende raziskovalne robotike.

Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams

FOREWORD

Robotics conference in the scope of the Information Society has been organized five times in the last six years and we are happy to be again a part of the multiconference after a year's break. The conference is a continuation of research robotics in Slovenija.

Even though many people still perceive robotics as science fiction, research robotics, industrial robotics and even consumer robotics have passed from the realm of fiction a long time ago. Many industrial processes today simply cannot be conceived without the use of robotic manipulators. The use of surgical and mobile service robots, which are about to enter human homes, are not a surprise anymore. As human imagination and wishes do not know any borders, research robotics is on the brink of a breakthrough, which will enable wide-spread use of multipurpose autonomous robotic household assistants. The development of such systems, which some consider harder than rocket science, requires cooperation between researchers and the exchange of idea and opinions, which is the reason the Robotics conference in the scope of the multiconference.

The conference proceedings contain researchers from the Department for Automatics, Biocybernetics and Robotics of Jožef Stefan Institute, and we are delighted to have attracted contributions of three research institutions in Serbia. We hope that the exchange of ideas will lead to future joint undertakings and will help to co-shape the trends of research robotics in the future.

Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams

ACTIVE STEREO 3-D POINT RECONSTRUCTION

Robert Bevec
Humanoid and Cognitive Robotics Lab
Department of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: robert.bevec@ijs.si

***Abstract*— Stereo reconstructed 3-D points serve as a basis for 3-D perception, tracking, object modeling and scene reconstruction. They are created using a pair of cameras by triangulating each 3-D point from its image projections. The camera's intrinsic and extrinsic parameters must be determined for the reconstruction to be accurate. However, there is an uncertainty in the measurement, which must be evaluated for further processing of the data. When one or both of the cameras are individually articulated, the geometrical relation between the stereo pair changes. It is necessary to model the motor system articulating the cameras to make 3-D reconstruction possible. There is also an unknown transformation between the camera and motor coordinate system, that must be determined by a calibration procedure. The calibration procedures involve fitting models to noisy data, which is reflected in the uncertainties of calibration parameters. These uncertainties are propagated to the final 3-D reconstruction. In this article we describe the articulated camera system - active stereo - calibration procedure and explain the uncertainty propagation from calibration to the final 3-D measurement.**

I. INTRODUCTION

Point clouds represent data sets in some coordinate system. When the data represent three dimensional positions, they serve as a basis for 3-D perception. Point clouds have been used widely for object modeling, object recognition, scene reconstruction and segmentation, pose estimation, object measuring and robot or vehicle guidance. They can be created by sensors using different techniques, e.g. measuring the time of reflected light, observing pattern deformations of structured light or triangulating a laser beam reflection. However, a point cloud can also be constructed without special sensors, using only a couple of cameras. A stereo camera systems reconstructs each 3-D point from its image projections. First the correct point correspondences must be found in each image. Assuming a pinhole camera model, the intersection of the projection rays, represents the reconstructed point [1]. In order to determine the intersection, the internal camera parameters and the parameters, defining the placement of the cameras in the world coordinate system or relative to each other, must be known. If camera lenses suffer radial distortion, the pinhole camera model can be extended by an empirical inverse model that corrects the distorted image coordinates [2]. Intrinsic, extrinsic and distortion parameters are estimated by camera calibration.

When one or both of the cameras are individually articulated, the relative arrangement of the cameras changes as the cameras move. Articulated cameras can be used to cover a greater area, in humanoid robotics they mimic the human ocular system [3] and they can be helpful for foveation setups [4]. However, it is necessary to model the motor system of the cameras to make 3-D reconstruction possible. Ude and Oztop [5] developed a calibration method for determining the transformation between the camera and motor coordinate systems, because the internal camera coordinate system cannot be aligned precisely with the end of the kinematic chain. They also describe how to realize 3-D point reconstruction as the cameras move.

The calibration procedures, however, involve fitting models to noisy data which is reflected in the uncertainties of calibration parameters. Noise occurs due to incorrect image point localization, imprecise calibration object, camera sensor noise, etc. The uncertainty of parameters is exhibited in the uncertainty of the final 3-D measurement. In previous works, there have been studies examining uncertainty propagation in stereo systems [6], [7], however, the cameras in those systems were not articulated.

In this article we describe the articulated camera system calibration procedure and explain the uncertainty propagation. We presume that the input noise is normally distributed and that the input-output transformation is well approximated by a linear transformation. We do not deal with the intrinsic errors, that are caused by sensor noise.

II. STEREO RECONSTRUCTION BASICS

A pinhole camera model maps a 3-D world point $\mathbf{P}$ to its 2-D projection in the image $\mathbf{p}$ as seen in Figure 1. The projection is described by a linear mapping described in Eq. (1), where λ is an arbitrary scale factor. The parameters u_{p} and v_{p} represent the principal point, α_{u} and α_{v} represent the horizontal and vertical focal length expressed in pixels, respectively and γ represents the skewness of the image axes. These parameters, called intrinsic parameters, are later referred by $\mathbf{p}_{\mathrm{int}} = [u_{\mathrm{p}}, v_{\mathrm{p}}, \alpha_{\mathrm{u}}, \alpha_{\mathrm{v}}, \gamma]$. $\mathbf{R}_{\mathrm{c}}$ and $\mathbf{t}_{\mathrm{c}}$ represent the rotation matrix and translation vector relating the camera and world reference systems. Since the rotation matrix has 9 elements, but only 3 DoF, we will later use the Rodrigues-vector representation for denoting rotation parameters $\mathbf{r}_{\mathrm{c}} \in$

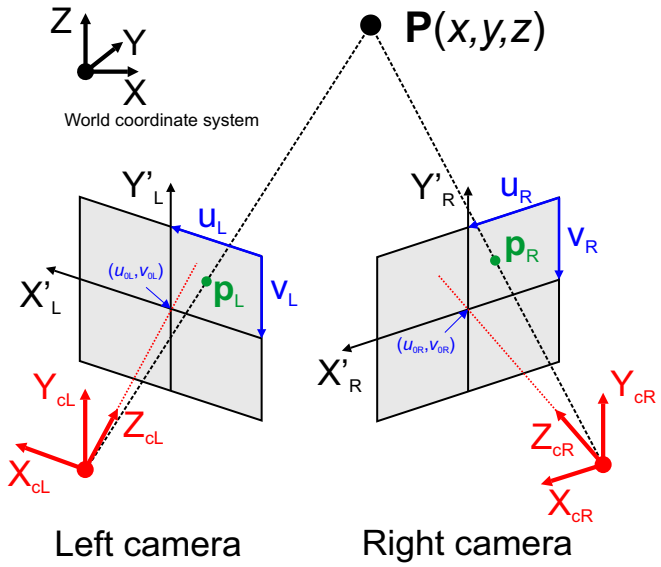


Fig. 1. Geometrical model of a stereo system.

$\Re^3$. The rotation and translation of the camera represent its extrinsic parameters $\mathbf{p}_{\text{ext}} = [\mathbf{r}_c, \mathbf{t}_c]$.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & \gamma & u_p & 0 \\ 0 & \alpha_v & v_p & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

Due to the distortion caused by the lens, we apply the empirical inverse model that corrects the image coordinates [2]. This introduces another set of calibration parameters $\mathbf{p}_{\text{dist}} = [k_{r1}, k_{r2}, k_{r3}, k_{t1}, k_{t2}]$ for mapping a world point to the image. This mapping becomes nonlinear $\varphi : \mathbf{P} \mapsto \hat{\mathbf{p}}$ and can be written in a general inhomogeneous form as

$$\hat{\mathbf{p}} = \varphi(\mathbf{p}_{\text{int}}, \mathbf{p}_{\text{ext}}, \mathbf{p}_{\text{dist}}, \mathbf{P}) \quad (2)$$

The 3-D reconstruction procedure evaluates the intersection between the lines of sight defined by the image point localizations. Let $\tilde{\mathbf{p}}_L$ and $\tilde{\mathbf{p}}_R$ be the projections found in each original image, respectively. These coordinates are distorted, therefore we first apply the inverse model that corrects the distortion. After that a well known technique called linear triangulation finds the solution by solving the overdetermined linear system with least-square technique [1]. We write the entire mapping in a general form

$$\psi(\widetilde{\mathbf{P}}, \mathbf{p}_{\text{intL}}, \mathbf{p}_{\text{extL}}, \mathbf{p}_{\text{distL}}, \tilde{\mathbf{p}}_L, \mathbf{p}_{\text{intR}}, \mathbf{p}_{\text{extR}}, \mathbf{p}_{\text{distR}}, \tilde{\mathbf{p}}_R) = 0, \quad (3)$$

where $\widetilde{\mathbf{P}}$ is the solution of the triangulation procedure and the other inputs are individual camera parameters and point localizations.

III. CALIBRATION

In order to reconstruct a 3-D point the parameters of the left and right camera have to be determined by a calibration algorithm. Although several stereo calibration algorithms exist, the common underlying idea behind them is similar. Acquire the left and right images of calibration points, whose relative or absolute position is known. Since planar calibration objects are easier to create than 3D ones, recent methods prefer using planar patterns. Unfortunately, from a single planar arrangement it is impossible to recover all the camera parameters. Zhang presents a solution to this problem by showing the cameras the same known pattern from different unknown orientations [8]. The system of equations for the calibration can be generally written as:

$$\xi(\mathbf{p}_{\text{stereo}}, \mathbf{c}) = 0 \quad (4)$$

The best static stereo system parameters $\mathbf{p}_{\text{stereo}} = [\mathbf{p}_{\text{intL}}, \mathbf{p}_{\text{distL}}, \mathbf{p}_{\text{intR}}, \mathbf{p}_{\text{distR}}, \mathbf{p}_{\text{LR}}]$ are found by minimizing the distance between the localized pattern positions in the left and right images $\mathbf{c} = [\mathbf{c}_1, ..., \mathbf{c}_N] = [u_1^{\text{L}}, v_1^{\text{L}}, u_1^{\text{R}}, v_1^{\text{R}}, ..., u_N^{\text{L}}, v_N^{\text{L}}, u_N^{\text{R}}, v_N^{\text{R}}]$ and the pattern 3-D positions projected back to the image. The calibration procedure returns $\mathbf{p}_{\text{LR}}$, which is the relative pose from one camera to another. Since the positions of the calibration points are not given in any specific coordinate system, it is assumed the world coordinate system is in one of the cameras, in this case the left camera. To apply the triangulation procedure in Eq. (3), $\mathbf{p}_{\text{extL}}$ becomes $[0, 0, 0, 0, 0, 0]$ and $\mathbf{p}_{\text{extR}} = \mathbf{p}_{\text{LR}}$ The application of this calibration procedure is as follows:

- Create a flat chessboard pattern
- Measure the chessboard square size
- Articulated cameras are put in a known static configuration
- Acquire left and right images of the chessboard at different orientations
- Extract chessboard corners in the images
- Input the data to the stereo calibration algorithm

In case of articulated cameras, it is necessary to model the motor system of the cameras to make 3-D reconstruction possible. As the cameras move, the relative arrangement of the cameras changes. However, it is very difficult to mount the cameras on the head so that the internal camera coordinate system would be aligned with the kinematic chain end effector. To calculate how the cameras move, we need to estimate the unknown transformation from the kinematic model coordinate system to the camera coordinate system. Ude and Oztop developed a method, where a calibration object is placed at a fixed location and acquired by the camera at different camera poses [5]. The end effector pose of the articulated system, where the camera is mounted, must be computed for each different snapshot of the calibration pattern. This is achieved by reading the motor joint values and considering the kinematic structure of the articulated system. The calibration pattern pose is also extracted from each snapshot, following the method in [8]. In Figure 2 we can see the relationship between these poses at two different snapshots.

Let $\mathbf{T}_{\text{k}}^j$ and $\mathbf{T}_{\text{c}}^j$, $j = 0, ..., n$, be the poses of the kinematic chain end effector in the camera motor base coordinate system and the calibration pattern pose in the camera coordinate system, respectively. $\mathbf{T}_{\text{fix}}$ is the unknown transformation we are looking for. Based on Figure 2, we can state the following

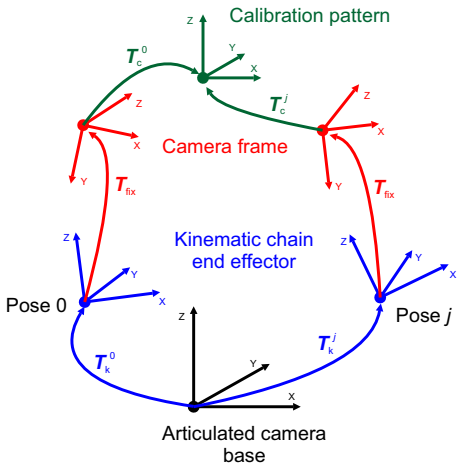


Fig. 2. Graphical representation of the kinematic chain and calibration object pose at 0-th and j-th snapshot in the active calibration procedure.

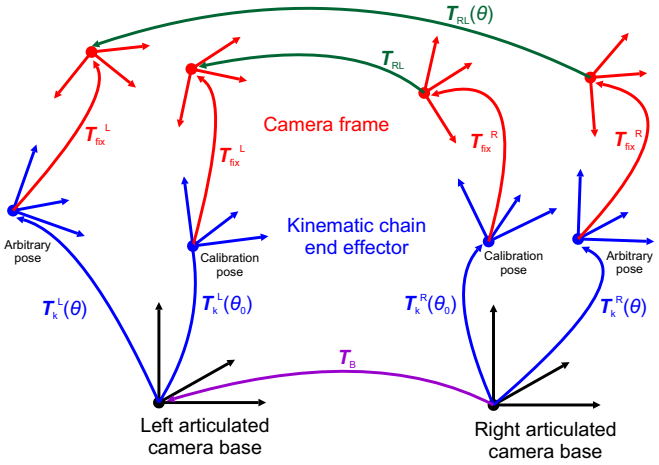


Fig. 3. The relative pose from one camera to another $\mathbf{T}_{\mathrm{RL}}(\theta)$ dependent on the joint values θ. This relation needs to be estimated for 3-D point reconstruction to be possible on an active stereo system.

for each $j, j = 1, ..., n$, if we denote $\mathbf{A}_j = \mathbf{T}_{\mathrm{k}}^{0\,-1}\mathbf{T}_{\mathrm{k}}^{j}$ and $\mathbf{B}_j = \mathbf{T}_{\mathrm{c}}^{0}\mathbf{T}_{\mathrm{c}}^{j\,-1}$

$$\mathbf{A}_j\mathbf{T}_{\mathrm{fix}} = \mathbf{T}_{\mathrm{fix}}\mathbf{B}_j \tag{5}$$

The equation system in Eq. 5 can be solved analytically bearing in mind that $\mathbf{A}_j, \mathbf{B}_j, \mathbf{T}_{\mathrm{fix}} \in \mathrm{SE}(3)$, $\mathrm{SE}(3)$ being the special Euclidean group of rigid body transformations. We repeat the solution derived in [5] for the subsequent sensitivity analysis.

The unknown rotation of $\mathbf{T}_{\mathrm{fix}} = \begin{bmatrix} \mathbf{R}_{\mathrm{fix}} & \mathbf{t}_{\mathrm{fix}} \\ 0 & 1 \end{bmatrix}$ is determined first

$$\mathbf{R}_{\mathrm{fix}} = (\mathbf{M}^T\mathbf{M})^{-1/2}\mathbf{M}^T,\ \mathbf{M} = \sum_{j=1}^{N}\mathbf{a}_j\mathbf{b}_j^T \tag{6}$$

which is then used to determine the translation $\mathbf{t}_{\mathrm{fix}}$ using a least-squares technique on the equation system

$$(\mathbf{I} - \mathbf{R}_{\mathbf{A}_j})\mathbf{t}_{\mathrm{fix}} = \mathbf{t}_{\mathbf{A}_j} - \mathbf{R}_{\mathrm{fix}}\mathbf{t}_{\mathbf{B}_j},\ j = 1, ..., n \tag{7}$$

$\mathbf{a}_j$ and $\mathbf{b}_j$ denote the Rodrigues vector representations of the rotation $\mathbf{R}_{\mathbf{A}_j}$ and $\mathbf{R}_{\mathbf{B}_j}$ from $\mathbf{A}_j$ and $\mathbf{B}_j$. $\mathbf{t}_{\mathbf{A}_j}$ and $\mathbf{t}_{\mathbf{B}_j}$ denote their translation part, respectively. The entire calibration can be generally written as:

$$\zeta(\mathbf{T}_{\mathrm{fix}}, \mathbf{T}_{\mathrm{K}}, \mathbf{T}_{\mathrm{C}}) = 0, \tag{8}$$

where $\mathbf{T}_{\mathrm{K}} = [\mathbf{T}_{\mathrm{k}}^0, ..., \mathbf{T}_{\mathrm{k}}^n]$ and $\mathbf{T}_{\mathrm{C}} = [\mathbf{T}_{\mathrm{c}}^0, ..., \mathbf{T}_{\mathrm{c}}^n]$. The application of the active stereo camera calibration procedure is as follows:

- Place the chessboard pattern at a fixed position
- Articulate the camera to an arbitrary position, providing the chessboard is still visible
- Extract the pose of the chessboard and the kinematic chain
- Repeat the previous two steps n-times
- Input the data to the active calibration algorithm

This procedure must be done for each articulated camera in the active stereo system to estimate its unknown transformations $\mathbf{T}_{\mathrm{fix}}$.

IV. ACTIVE STEREO RECONSTRUCTION

The relative pose from one camera to another $\mathbf{T}_{\mathrm{RL}}$, returned by the static calibration is not constant on an active system. It needs to be estimated as the cameras move. After the kinematic model of the articulated cameras has been identified and the active stereo system calibrated, we have all the necessary information for the estimation.

Let θ be the vector representing joint values of the left and right articulated systems. $\mathbf{T}_{\mathrm{RL}}(\theta)$ now depends on the joint values. Since the static calibration was done in a known configuration θ_0, $\mathbf{T}_{\mathrm{RL}}(\theta_0) = \mathbf{T}_{\mathrm{RL}}$. From Figure 3 it is clear that we can calculate the transformation $\mathbf{T}_{\mathrm{B}}$ between the articulated camera bases, which is constant.

$$\mathbf{T}_{\mathrm{B}} = \mathbf{T}_{\mathrm{k}}^{R}(\theta_0)\,\mathbf{T}_{\mathrm{fix}}^{R}\,\mathbf{T}_{\mathrm{RL}}\,\mathbf{T}_{\mathrm{fix}}^{L\,-1}\,\mathbf{T}_{\mathrm{k}}^{L}(\theta_0)^{-1} \tag{9}$$

With $\mathbf{T}_{\mathrm{B}}$ known the estimation of the relative pose from the right camera to the left is

$$\mathbf{T}_{\mathrm{RL}}(\theta) = \mathbf{T}_{\mathrm{fix}}^{R\,-1}\,\mathbf{T}_{\mathrm{k}}^{R}(\theta)^{-1}\,\mathbf{T}_{\mathrm{B}}\,\mathbf{T}_{\mathrm{k}}^{L}(\theta)\,\mathbf{T}_{\mathrm{fix}}^{L} \tag{10}$$

With the relative pose between the cameras known, the reconstruction procedure is trivial. By placing the world coordinate system in the left camera, $\mathbf{p}_{\mathrm{extL}}$ becomes $[0, 0, 0, 0, 0, 0]$ and $\mathbf{p}_{\mathrm{extR}}$ equals the Rodrigues rotation vector and the translation vector corresponding to $\mathbf{T}_{\mathrm{RL}}(\theta)$. Applying Eq. (3) then returns the 3-D reconstructed point in the left camera coordinate system.

V. UNCERTAINTY ANALYSIS

When uncertain data are used as inputs to an algorithm, the standard uncertainty of algorithms results can be evaluated with an analytical approach, developing analytical relationships describing how uncertainty propagates through the algorithm from inputs to outputs.

The input parameters for the static stereo calibration in Eq. (4) are the image localizations of the calibration pattern. The data for the calibration is noisy, due to incorrect image point localization, imprecise calibration object and camera

sensor noise. We assume sensor noise is negligible and the calibration object is accurate, which is a viable assumption for modern cameras and printed planar patterns. If the distribution of image localization errors is a Gaussian with zero mean, then the distance between the localized pattern positions and the reprojected pattern 3-D positions represents a good measure of this noise. We denote σ_c as the standard deviation of the static calibration input noise

$$\sigma_c^2 = \text{var}(\hat{\mathbf{c}} - \mathbf{c}), \tag{11}$$

where $\hat{\mathbf{c}}$ are the reprojected pattern 3-D positions following Eq. (2).

When input quantities are uncertain, the uncertainty of the output u can be estimated with [9]:

$$u^2 = \mathbf{J}\mathbf{\Lambda}\mathbf{J}^{\mathrm{T}}, \tag{12}$$

where $\mathbf{\Lambda}$ is the input covariance matrix and $\mathbf{J}$ is the partial derivatives matrix of the output function with respect to the inputs. From the implicit Eq. (4) we can therefore derive:

$$\mathbf{J}_s\mathbf{\Lambda}_s\mathbf{J}_s^{\mathrm{T}} = \mathbf{J}_c\mathbf{\Lambda}_c\mathbf{J}_c^{\mathrm{T}}, \tag{13}$$

where $\mathbf{\Lambda}_c$ is a diagonal matrix $\mathbf{\Lambda}_c = \text{diag}\{\mathbf{\Lambda}_{c1}, ...\mathbf{\Lambda}_{cN}\}$. Since the input quantities are uncorrelated each $\mathbf{\Lambda}_{ci}$ is diagonal $\mathbf{\Lambda}_{ci} = \text{diag}\{\sigma_c, \sigma_c, \sigma_c, \sigma_c\}$. $\mathbf{J}_c$ is the Jacobian matrix of derivatives ξ with respect to the input parameters $u^{\mathrm{L}}, v^{\mathrm{L}}, u^{\mathrm{R}}, v^{\mathrm{R}}$ evaluated at each calibration value. $\mathbf{J}_c = \text{diag}\{\mathbf{J}_{c1}, ...\mathbf{J}_{cN}\}$.

$$\mathbf{J}_{ci} = \left[\begin{array}{cccc} \frac{\partial \xi}{\partial u^{\mathrm{L}}}(\mathbf{c}_i) & \frac{\partial \xi}{\partial v^{\mathrm{L}}}(\mathbf{c}_i) & \frac{\partial \xi}{\partial u^{\mathrm{R}}}(\mathbf{c}_i) & \frac{\partial \xi}{\partial v^{\mathrm{R}}}(\mathbf{c}_i) \end{array} \right] \tag{14}$$

$\mathbf{J}_s$ is the Jacobian matrix of derivatives ξ with respect to the 26 output parameters in $[\mathbf{p}_{\text{intL}}, \mathbf{p}_{\text{distL}}, \mathbf{p}_{\text{intR}}, \mathbf{p}_{\text{distR}}, \mathbf{p}_{\text{LR}}]$, $\mathbf{J}_s = [\mathbf{J}_{s1}^{\mathrm{T}}, ..., \mathbf{J}_{sN}^{\mathrm{T}}]^{\mathrm{T}}$ evaluated at each calibration value $\mathbf{c}_i, i = 1, ..., N$.

$$\mathbf{J}_{si} = \left[\begin{array}{cccc} \frac{\partial \xi}{\partial u_{\mathrm{p}}^{\mathrm{L}}}(\mathbf{c}_i) & \frac{\partial \xi}{\partial v_{\mathrm{p}}^{\mathrm{L}}}(\mathbf{c}_i) & ... & \frac{\partial \xi}{\partial t_{\mathrm{LR}}^{z}}(\mathbf{c}_i) \end{array} \right] \tag{15}$$

Finally the covariance matrix of the static calibration output can be calculated:

$$\mathbf{\Lambda}_s = \mathbf{J}_s^{\Delta}\mathbf{J}_c\mathbf{\Lambda}_c\mathbf{J}_c^{\mathrm{T}}\mathbf{J}_s^{T\,\Delta}, \tag{16}$$

with the variance of the parameters in the diagonal of $\mathbf{\Lambda}_s$ and $\mathbf{J}^{\Delta} = (\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}$ is the pseudoinverse matrix of $\mathbf{J}$.

The camera parameters are used in the motor-camera calibration described in Section III, where their uncertainty propagates into the estimated transformation between the camera and motor coordinate systems in Eq. (8). The uncertainty of the kinematic poses $\mathbf{T}_{\mathrm{K}}$ also influences this estimation. The covariance matrices of the kinematic poses $\{\Lambda_{k0}, ..., \Lambda_{kn}\}$ and the chessboard poses $\{\Lambda_{c0}, ..., \Lambda_{cn}\}$ represent the elements of the input covariance matrix $\Lambda_{kc} = \text{diag}\{\Lambda_{k0}, \Lambda_{c0}, ...\}$. The covariance of the output is calculated following the same procedure as in Eq. (16). Jacobi matrices of partial derivatives ζ with respect to the inputs and outputs are calculated and the covariance Λ_{fix} of the transformation $\mathbf{T}_{\text{fix}}$ estimated.

With the transformation matrices for the left and right camera including their uncertainties estimated, the uncertainty of the pose from the left camera to the right can be computed following Eq. (10), where the uncertainty of the kinematic pose is included again.

When the uncertainty of the pose from the right to the left camera is estimated, the uncertainty of the final 3-D measurement can be computed with the image correspondences $\mathbf{p}_{\mathrm{L}}$, $\mathbf{p}_{\mathrm{R}}$ and their uncertainties σ_{L} and σ_{R}:

$$\mathbf{\Lambda} = \mathbf{J}^{\Delta}\mathbf{J}_{3\mathrm{D}}\mathbf{\Lambda}_{3\mathrm{D}}\mathbf{J}_{3\mathrm{D}}^{\mathrm{T}}\mathbf{J}^{T\,\Delta},$$

$$\mathbf{J} = \left[\begin{array}{ccc} \frac{\partial \psi}{\partial x}(\rho) & \frac{\partial \psi}{\partial y}(\rho) & \frac{\partial \psi}{\partial z}(\rho) \end{array} \right]$$

$$\mathbf{J}_{3\mathrm{D}} = \left[\begin{array}{ccccccc} \frac{\partial \psi}{\partial u_{\mathrm{p}}^{\mathrm{L}}}(\rho) & ... & \frac{\partial \psi}{\partial u_{\mathrm{L}}}(\rho) & \frac{\partial \psi}{\partial v_{\mathrm{L}}}(\rho) & \frac{\partial \psi}{\partial u_{\mathrm{R}}}(\rho) & \frac{\partial \psi}{\partial v_{\mathrm{R}}}(\rho) \end{array} \right] \tag{17}$$

, evaluated at $\rho = [\mathbf{p}_{\text{stereo}}, \mathbf{p}_{\mathrm{L}}, \mathbf{p}_{\mathrm{R}}]$. $\mathbf{\Lambda}_{3\mathrm{D}} = \text{diag}(\mathbf{\Lambda}_{\mathrm{A}}, \sigma_{\mathrm{L}}, \sigma_{\mathrm{L}}, \sigma_{\mathrm{R}}, \sigma_{\mathrm{R}})$

VI. CONCLUSIONS

The 3-D reconstruction reliability on an active stereo system depends greatly on several parameters. The calibration procedure is crucial, since it sets a minimum uncertainty of the reconstruction, even when the image correspondences are perfect. The calibration procedure relates uncertainties from two different aspects of the system. The kinematic model of the articulated system must be appropriate, since uncertainties of the joint values and the kinematic model end up in the end effector pose estimation. The other aspect is the camera calibration procedure, which estimates the camera parameters. We have analytically shown how the uncertainty are propagated to the final 3-D measurement. From these relations, it is possible to evaluate which aspect brings greater uncertainties and, if necessary, improve it.

REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, IEEE Comput. Soc.

[3] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati, "Active vision for sociable robots," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 443–453, 2001.

[4] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann, "The Karlsruhe Humanoid Head," in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 447–453, IEEE, 2008.

[5] A. Ude and E. Oztop, "Active 3-D vision on a humanoid head," in *2009 International Conference on Advanced Robotics*, pp. 1–6, 2009.

[6] J. Chen, Z. Ding, and F. Yuan, "Theoretical Uncertainty Evaluation of Stereo Reconstruction," in *International Conference on Bioinformatics and Biomedical Engineering*, no. 2, pp. 2378–2381, IEEE, May 2008.

[7] G. Di Leo, C. Liguori, and a. Paolillo, "Covariance Propagation for the Uncertainty Estimation in Stereo Vision," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 1664–1673, May 2011.

[8] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[9] M. G. Cox and P. M. Harris, "Software Support for Metrology Best Practice Guide No. 6 Uncertainty Evaluation," tech. rep., National Physical Laboratory, y, Teddington, UK, 2006.

MODULATION OF COMBINED DISCRETE-PERIODIC DYNAMIC MOVEMENT PRIMITIVES WITH ITERATIVE LEARNING CONTROL AND REPETITIVE CONTROL

Andrej Gams
Humanoid and Cognitive Robotics Lab
Department of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: andrej.gams@ijs.si

ABSTRACT

The framework of dynamic movement primitives (DMPs) contains many properties, which are favorable for the execution of robotic trajectories. These include such properties as indirect dependence on time, response to perturbations, and the ability to easily modulate the given trajectories. Applicability has been shown for both discrete point-to-point movements as well as for periodic movements. Thus far only one paper has discussed encoding combined transient discrete motion, followed by periodic motion, both encoded in a single movement primitive with a common canonical system. In this paper we provide the details for the execution of such DMP which combines transient motion as well as periodic motion and briefly discuss the means of modulating the trajectories of motion in both parts, i.e. in the transient – discrete – part using iterative learning control and in periodic part using repetitive control. Simulation results provide an assessment of the applicability of the system for robotic tasks which include transient motion and infinite periodic motion.

1. INTRODUCTION

Dynamic movement primitives (DMPs), first introduced by Ijspeert et al. [1], model attractor behaviors of autonomous nonlinear dynamical systems with the help of statistical learning techniques and thus provide the means to encode a trajectory. They comprise a set of differential equations that can compactly represent control policies, while their attractor landscapes can be adapted by only changing a few parameters. The latter can be exploited in several different ways, for example, for reinforcement learning [2], [3] and statistical generalization [4]. They can even be used for combining of separate trajectories in a dynamic way [5].

In a recent paper, Ernesti et al. [6] have shown how the framework can be extended to incorporate transient, discrete point-to-point motion, which continuously transits into periodic, i. e. infinite motion. Both are encoded in a single, extended DMP, with a common canonical system. The combination of transient and periodic motion encoded in a single system can be used for generation of periodic movements, where the start of the motion is not exactly the same as the following periodic motion. Periodic motions must be started in a non-periodic way before the repeating pattern comes into play [6]. Starting the motion cannot be a mere reproduction of part of the periodic pattern. For example, when one is standing and then starts walking, the first step is different from the other following steps. Similar applies to juggling, where the balls have to be thrown up in the air at first before one can juggle the balls in a periodic motion.

While DMPs are trivial to apply for the reproduction of motion, including for the case of the transient motion, the execution of the motion on the robot might not achieve the same results due to different kinematic and dynamic properties of the robot and the human demonstrator. Some means of adaptation of the DMP based on feedback need to be applied in order to change the motion to achieve the desired effect.

The DMPs themselves allow for easy modulation in duration or frequency, the goal and center of periodic motion, and the amplitude [7]. Gams et al. have also introduced approaches to modulate DMPs based on force feedback [8]. For the iterative case, it was shown that in a few iterations, the DMP can be modulated to practically cancel out any error of motion using iterative learning control (ILC) [8]. Similar was shown for periodic motion using repetitive control (RC) [9]. In the case of encoding both transient motion and periodic motion, we can combine these two learning techniques to achieve desired motion in both parts. The initial, transient motion is in several iterations updated using ILC, while RC modifies the periodic part. The latter can be updated in several iterations while the former has to be restated several times as ILC demands that the initial conditions are reset for every iteration [10].

The paper is organized as follows. In the next section

we introduce the DMP encoding both transient and periodic motion, as was described by Ernesti et al. [6]. Section III shows how we can modify the motion using ILC and RC while Section IV gives results. A concluding section follows at the end.

2. DMP FORMULATION

In this section we give a brief recap of the formulation of the DMP as was introduced by Ernesti et al. [6].

A DMP is composed of a canonical system and a transformation system. The canonical system provides the phase. Weighted kernel functions are anchored at this phase. Their weighted combination is added to a second order differential equation – the transformation system – at the level of the second derivative, i.e. at the acceleration level. The transformation system is for the case of both transient and periodic system given by

$$\dot{z} = \Omega\left(\alpha_z\left(\beta_z\left(g-y\right)-z\right)+f(\phi,r)\right), \quad (1)$$

$$\dot{y} = \Omega z. \quad (2)$$

The output of the differential system equations is the reference of motion of the robot. Details on DMPs are discussed in [11].

We can see that the nonlinear part $f(\phi, r)$ depends on two variables, which are both the output of the canonical system. This is the difference from the discrete DMP, where the canonical system is given by a differential equation for exponential decay, which starts at 1 [11]. For periodic systems it is usually a simple phase oscillator, where the phase oscillates between 0 and 2π [7].

In the novel DMP encoding of both transient and later periodic motion, it is a limit cycle, but the initial conditions are not on the limit cycle. Therefore the phase depends on both the distance from the center of the phase plane (r) and the angle ϕ . The canonical system is given by

$$\dot{\phi} = \Omega \quad (3)$$

$$\dot{r} = \eta(\mu^{\alpha} - r^{\alpha})r^{\beta} \quad (4)$$

$$\phi(0) = \phi_0,\ r(0) = r_0. \quad (5)$$

Here μ is the radius of the limit cycle (usually 1) and $\alpha = 1/6$, $\beta = 1/1000$, $\eta = 35$ are constants, here the same as in [6], but can be set differently.

The initial conditions r_0 and ϕ_0 determine the phase for the transient part. A unique solution of the canonical system is given for any pair of r and ϕ, and the values can be calculated based on the transient time. To calculate it one integrates with negative dt, starting from the desired values at the limit cycle (r_d, ϕ_d), back to the initial values r_0 and ϕ_0. The desired values at the limit cycle determine at what point in the limit cycle the transient motion is finished.

Figure 1 shows the output of a canonical system for the initial conditions determined by the desired values of r and ϕ where the transient response fades out. The desired values were $r_d = 1.2 = \mu_1$ and $\phi_d = 0$. The figure also shows the locations and the width of the kernel functions for the discrete part, as well as the sections of the limit cycle where separate kernels of the periodic part are used for trajectory reconstruction.

The nonlinear part f is defined as

$$f(\phi,r) = \frac{\Sigma_{j=1}^{M}\psi_j(\phi,r)w_j + \Sigma_{i=1}^{N}\varphi_i(\phi,r)w_i}{\Sigma_{j=1}^{M}\psi_j(\phi,r) + \Sigma_{i=1}^{N}\varphi_i(\phi,r)} \quad (6)$$

where w_i, w_j represent the weights which are adjusted to fit the desired trajectory.

The canonical system provides the phase for both the transient part and for the periodic part. At the beginning of the motion, the trajectory should only be governed by the transient part, and later only by the periodic part. At the transition of one to the other, it is governed by both, blended together.

For the transient part, we therefore write

$$\psi_j(\phi,r) = a(r)b_j \quad (7)$$

where the function to ensure only the transient part is observed is defined as

$$a(r) = \begin{cases} exp(-\tilde{v}(r-\mu_2)^k) & r \in (0,\mu_2) \\ 1 & r \in [\mu_2,\infty) \end{cases} \quad (8)$$

The kernel function itself is given by

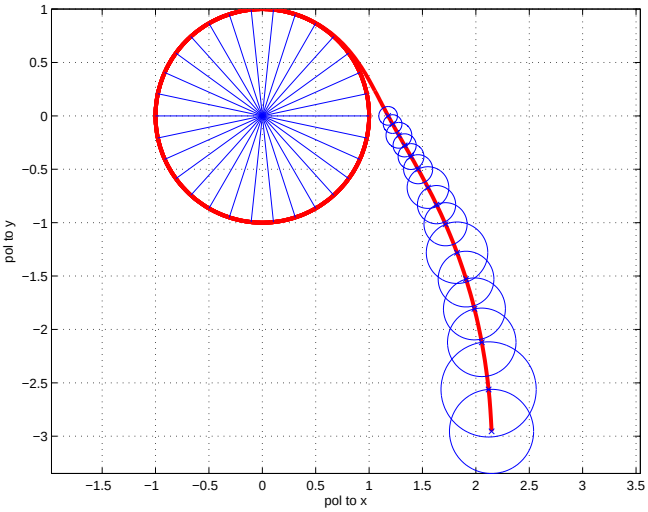


Fig. 1. The evolution of the canonical system, given in red. The desired end of the transient part, where it completely fades out at $r = \mu_1$, in this case at $r = 1.2$, were calculated from the desired location on the limit cycle, i.e. where it reaches μ_1. These were set to $r = \mu_1$ and $\phi = 0$ with a transient time of 0.6s. The blue circles give the kernels of the transient part and the kernels of the periodic part, which are defined by the phase of the limit cycle.

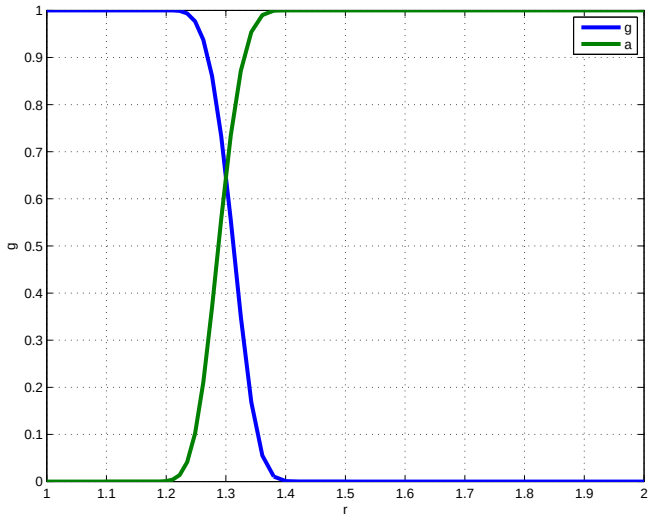


Fig. 2. The blending of the transient and the periodic part is governed by the terms a and g.

$$b_j = \exp(-\sigma_j(||\mathbf{pol}(\phi, r) - q_j||)^2), \tag{9}$$

with $\mathbf{pol}(\phi, r) = (r\cos\phi, r\sin\phi)$ the polar coordinate mapping in the phase plane and $\sigma_j = \frac{2}{||q_{j+1}-q_j||^2}$, $\sigma_N = \sigma_{N-1}$. $\mathbf{q}$ defines the locations of the discrete kernel functions in the phase plane. It is populated so that discrete kernels are symmetrically distributed over time.

For the periodic part, on the other hand, we write

$$\varphi_i(\phi, r) = g(r)h_i(\phi), \tag{10}$$

where h_i is 2π periodic and encodes the periodic motion. We use

$$h_i(\phi) = \exp(v_i(cos(\phi - c_i) - 1)) \tag{11}$$

$$g(r) = \begin{cases} 1 & r \in (0, \mu_1) \\ exp(-\tilde{v}(r-\mu_1)^k) & r \in \mu_1, \infty) \end{cases} \tag{12}$$

Here v_i determines the width of kernel functions, $k = 4$ as in [6] and

$$c_i := (i-1)2\pi/n. \tag{13}$$

The blending of the transient and the periodic part takes place between μ_2 and μ_1, which is depicted in Fig 2.

To reproduce a demonstrated trajectory, given by y_{demo}, we manually select the transient and periodic parts. The duration of the transient part determines the initial values of the canonical system (ϕ_0, r_0). To learn the weights we create a target for fitting, which is derived from (1) into

$$f_{targ}(\phi, r) = \frac{1}{\Omega^2}\ddot{y}_{ref} - \alpha_z\left(\beta_z\left(g - y_{ref}\right) - \frac{1}{\Omega}\dot{y}_{ref}\right), \tag{14}$$

The weights are fitted using a locally weighted regression, see [4] for details.

Note that all the weights are calculated at once, i.e. for both the transient and the periodic motions and that they do not have to be calculated separately.

3. DMP ADAPTATION

We implemented adaptation of the DMP using iterative learning control for the transient, i.e. discrete part, and repetitive control for the periodic part. Similar as in the formulation of the DMP, we multiply their effects on the DMP with a and g terms.

The DMP was modulated with a coupling term at the velocity level, changing (2) into

$$\dot{y} = \Omega\left(z + C\right). \tag{15}$$

Here C represents a coupling term, composed of feedback (c_{fb}) and feed-forward (c_{ff}) terms

$$C = c_{\mathrm{fb}} + c_{\mathrm{ff}}. \tag{16}$$

The feedback term is defined by

$$c_{\mathrm{fb}} = k\left(F_{\mathrm{des}} - F_{\mathrm{act}}\right), \tag{17}$$

where k is a positive constant, F_{des} is the desired force trajectory and F_{act} is the real, measured force. The desired force can be any force trajectory. A virtual force can also be used, defined for example by a virtual spring between the position of the robot y_{r1} and an arbitrary object o

$$F_{\mathrm{act,v}} = k_{\mathrm{env}}(o - y_{r1}). \tag{18}$$

with k_{env} defining the stiffness of the virtual spring.

The feedforward term is defined for ILC, as suggested by [10]

$$c_{ff,i} = d\,e_i + F_{c,i}, \tag{19}$$

$$F_{c,i} = Q(F_{c,i-1} + L\,c\,e_{i-1}) \tag{20}$$

$$e_i = F_d - F_i, \tag{21}$$

where index i denotes the i-th epoch, d is the force gain, e_i is the coupling force error calculated from the difference of the desired coupling force F_d and the measured coupling force $F_i = F_{1,i} - F_{2,i}$, $F_{c,i}$ is the learned coupling force term, and Q and L are positive scalars.

For repetitive control, the equations are very similar, but we are not dealing with epochs but with periods. At the start of the next period we use the the error signal and coupling signal, which are delayed for one period. For details on repetitive control and the relation towards ILC check Wang et al. [12].

4. RESULTS

In this section we show simulated results of adapting a periodic motion which includes a transient motion, to a different motion of the same kind.

Figure 3 shows the original signal, recorded from a demonstration and encoded in a DMP, in the top plot. The

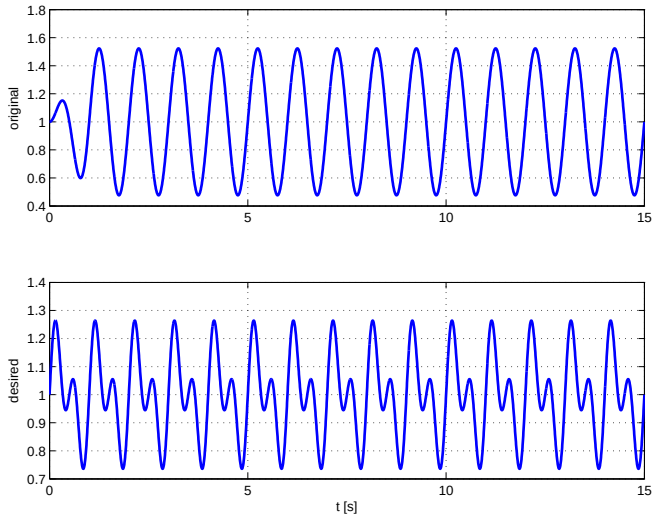


Fig. 3. The original signal in the top plot and the desired signal in the bottom plot.

bottom plot shows the desired signal, which could not be achieved through demonstration. We can see that the signals differ considerably, but have the same frequency, which is a necessary condition.

Figure 4 shows the error of adaptation of a periodic signal with a transient motion, to the desired motion. We can see that the error in the first epoch is considerably larger than in the second attemp – epoch. In the last epoch the error is practically gone, apart from the initial moments of the experiment.

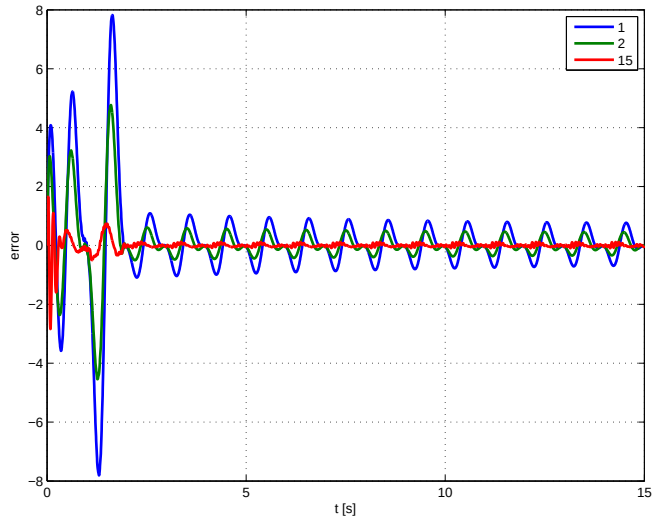


Fig. 4. The error between the reproduced and desired signal in the first, second and the last, 15th epoch of repetition of the periodic task with a transient motion.

5. CONCLUSION

In this paper we augmented the combined discrete periodic dynamic movement primitive, developed by Ernesti et al. [6], with iterative learning control and repetitive control coupling parts for the discrete – transient part and the periodic infinite part, respectively. We can see that a DMP can encode both transient and periodic parts in a single system, where the phase is two-dimensional. We can also modulate the DMP the same as modulating a discrete or a periodic DMP, by introducing a coupling term at the velocity level, and using appropriate learning type control to learn the correct coupling values over a few periods.

The system is applicable for any periodic task which requires a different initial motion. An example of such is walking, where the first step is different. When appropriate cost functions are given, we can use them to adapt the motion to the desired shape and form. In the future we will combine the system with varying frequencies and external conditions, such as maintaining the stability of walking.

REFERENCES

[1] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, Washington, DC, 2002, pp. 1398–1403.

[2] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, pp. 682–697, 2008.

[3] J. Kober and J. Peters, "Learning motor primitives for robotics," in *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009, pp. 2112–2118.

[4] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[5] B. Nemec and A. Ude, "Action sequencing using dynamic movement primitives," *Robotica*, vol. 30, no. 05, pp. 837–846, 2012.

[6] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal, "Encoding of periodic and their transient motions by a single dynamic movement primitive," in *2012 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 57–64.

[7] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous Robots*, vol. 27, no. 1, pp. 3–23, 2009.

[8] A. Gams, B. Nemec, A. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, 2014, to appear.

[9] A. Gams, J. van den Kieboom, M. Vespignani, L. Guyot, A. Ude, and A. Ijspeert, "Rich periodic motor skills on humanoid robots: Riding the pedal racer," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[10] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, june 2006.

[11] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[12] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589 – 1600, 2009.

MECHANICAL AND CONTROL HARDWARE DESIGN OF HUMANOID ROBOT MARKO

Borovac Branislav, Raković Mirko, Savić Srđan, Nikolić Milutin, Aleksandar Batinica
Faculty of Technical Sciences
University of Novi Sad
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
e-mail: borovac@uns.ac.rs

ABSTRACT

Robot MARKO has been developed to participate in the therapy with children suffering from cerebral palsy. The robot should assist doctors in process of habilitation, by motivating children to spend more time exercising. The tasks assigned to the robot will involve free-space motion of arms and legs (gross motor skill exercises) as well as constrained motion due to contact with objects from its environment (fine motor skill exercises). The robot has head, two arms, two legs and sitting on a mobile platform. The control hardware consists of central control unit and distributed motion control units. The central unit is conceived as a higher level cognitive based controller which integrates all the subsystems including, motion control subsystems, vision subsystem, automatic speech recognition and text to speech subsystem.

1 INTRODUCTION

The development of rehabilitation and assistive robotics is rapidly progressing. For assistive robots it is most important to ensure that the robots act safely in human environment. Since these robots are designed to operate in human's living and working environment, robots appearance and behaviour should be such to be adequate for humans, i.e. they have to be socially acceptable. Some examples of assistive robots are robot assistant for elderly population "TWENDY-ONE" [1], and "Bandit", assistive robot for post-stroke rehabilitation, [2].

A lot of research has been conducted for the application of social robots in therapy of children with autism [3], [4], but only recently some results of robot application in cerebral palsy therapy have been reported [5], [6]. In [5] humanoid robot "KineTron" is presented, which acts like a coach to encourage the patient during the cerebral palsy therapy. It has nine predefined movement scenarios combined with voice and music. More advanced solution is the robot "Ursus" that acts as a child's playmate in a game scenario that combines real robot and virtual reality. Showing calmly the correct movements with his arms, talking about interesting matters for the child, playing music and projecting pictures, videos and augmented reality games on an external screen, are some of the resources that "Ursus" pulls out to capture the child's attention and interest [6].

2 MECHANICAL DESIGN OF ROBOT MARKO

Robot MARKO is designed as humanoid with two legs, two arms, torso and head [7]. It will be placed on the horse-like mobile platform with differential drive for motion in its environment (Fig. 1). Robot anthropomorphic structure is determined by its intended application. The robot will have a central cognitive system, which will integrate a dialog management system (with speech recognition and synthesis), stereo vision and motor skills. Having in mind importance of propper and adequate communication with patient for therapy success, the robot will be able to position propperly itself by detecting the patient, recognize what patient is doing and communicate with him in a most natural way. The experiments of integrated system including robot, stereo vision, speech recognition and synthesis system with cognitive dialog management system have already been realized [8].

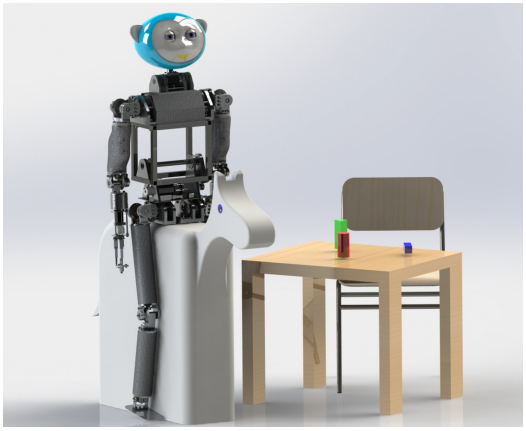


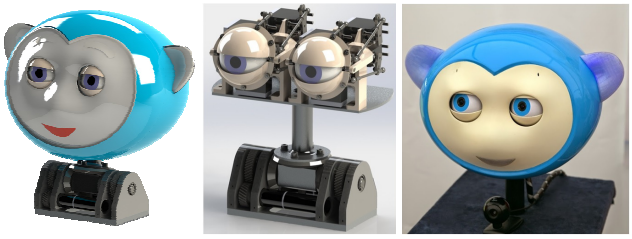
Figure 1: *Overall look of the humanoid robot MARKO*

Robot MARKO has 33 DOFs without hands and mobile platform. Most of the robot's arms and legs joints are actuated with brushless DC motors with integrated both incremental and 13 bit single turn and aditional 12 bit multi turn absolute encoders. MARKO has a pair of cameras in its

eyeballs which will be used only to detect a child's arms movements and overall position (with a stereo vision) and to provide switching to a compliant control mode at the right moment.

2.1 Head with neck

Robot neck has 2 DOFs and it is designed as differential mechanism with bevel gears. The movements that can be performed are flexion/extension (yes gesture) and rotation around vertical axis (no gesture). Eyes mechanism has 4 DOFs. Each eye can move upward/downward and to the left and right. Also upper and lower eyelids are independently actuated. Robot's face is manufactured on a 3D printer. CAD model of robot MARKO's head, mechanical design of robot's neck with eyes mechanism and manufactured head are shown in Fig. 2.



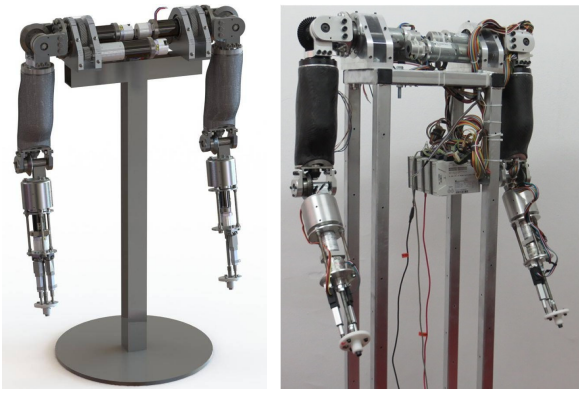
a) b) c)

Figure 2: *MARKO's head a) 3D model, b) mechanical design of neck and eyes, c) realized head*

2.2 Arms

For arms design, kinematic structure with 7 DOFs, closely approximating human arm, has been adopted. In Fig. 3 CAD model and arms prototype of are shown. Joint ranges, links dimensions, as well as their masses and position of centres of masses are designed to match human arm parameters as close as possible [7]. Specified limitations for maximum angular velocity and joints acceleration are ω=1,5 rad/s and α=7,5 rad/s^2, respectively. Maximum expected payload is 300g.

Shoulder joint has 3 DOFs and is approximated with three rotational joints whose axis of rotation intersect in one point.



a) b)

Figure 3: *MARKO's arms a) 3D model, b) realization of robot's arms*

Actuators for first and second DOF are inside the robot torso to reduce the masses and moments of inertia of the arm movable parts. All DOFs of shoulder are actuated with brushless DC motors from Dunkermotoren BG series, with integrated multi-turn digital absolute and incremental encoders to measure joint positions and velocities. Elbow joint has 1 DOF. For actuation of this joint Dunkermotoren BG32 brushless DC motor is used. Position and velocity measurement is implemented same way as in shoulder joint. Wrist joint has 3 DOFs whose axis of rotation intersect in one point. First wrist DOF (rotation about longitudinal axis) is actuated with the same motor as elbow and position and velocity measurement is implemented same way as in previous joints. The other two DOFs are actuated with linear actuators.

2.3 Legs

Robot legs have 4 DOFs. Hip joint has 2 DOFs, and it can perform flexion/extension and lateral flexion/extension movements. There is 1 DOF in the knee joint and 1 DOF in the ankle joint.

Figure 4: *3D model of MARKO's leg*

Legs have a extremely simple design. To achieve to be as light as possible carbon fiber tubes will be used for legs segments. Plastic covers, having propper shape will provide the final appearance of robot legs.

2 CONTROL HARDWARE DESIGN

Beside the body parts and motors for driving the joints, the robot consists of additional subsystems: motion control subsystem, vision subsystem, automatic speech recognition (ASR) together with text to speech (TTS) subsystem. All the subsystems are integrated in higher level cognitive control system [9].

The role of the vision subsystem is to detect the objects on the table that are used for fine motor excercises, to detect and track human's face when someone is standing in robot's immediate environment. Possible task to be also assigned to visual system is to realize and eveluate body movements performed by patient during exercies[1].

[1] It is also possible that this task will be assigned to external Kinect system.

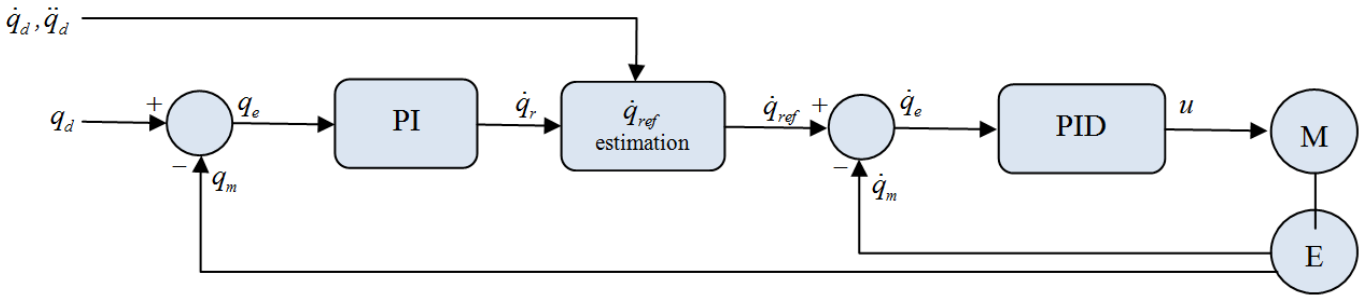


Figure 5. *Block diagram of cascade position-velocity control loop*

ASR and TTS are part of cognitive based dialog management system that is enabling a two way verbal communication with the users.

The motion control subsystem is designed as distributed with peripheral motion control boards. Each arm will have three, each leg will have two and head will have one peripheral board.

2.1 Controling the robots arms and legs

For motion control of motors that are driving joints in arms and legs, the peripheral board is designed that can control two brushless motors or one brushless motor and two servos. Microcotroler used for this peripheral boards is Atmel's Xmega128a1 controler.

Curently the position-velocity controler for each brushless DC motor is developed. Block diagram of control algorithm is given in Fig. 5. Variables q_d, $\dot{q}_d$ and $\ddot{q}_d$ are desired values of angle, angular valocity and acceleration. The q_m and $\dot{q}_m$ are measured angle and angular velocity. The measurements are obtained via serial synchroneous interface, and the recieved 25-bit position of the rotor shaft is represented in Gray code. For position-velocity, the PI+PID controler is addopted. The role of the PI controler is to determine the required andular velocity $\dot{q}_r$. In order to generate trapezoidal profile of angular velocity, the block for estimation of reference angular velocity is placed before PID velocity controler.

2.2 Controling the robots head

The control electronics for the head of the humanoid robot MARKO was intended to control the eyes, eyelids, eyebrows, ears and the mouth of the robot. The robot has two eyes, each having two DOFs, and two eyelids, each having one DOF. The eyebrows have three DOF. One common DOF, that moves the eyebrows up and down, and one on each eyebrow that rotates it. The total amount of DOF in the head of the robot is eleven. Each DOF is actuated by a servo.

All the servos excluding the ones for rotating the eyebrows are Modelcraft's MC-620 MG-T. The ones for rotating the brows are Tower Pro's SG 50. Each of these servos are controlled by a single PWM signal. The robot has two ears and a mouth. Each of these is equipped with three RGB LEDs.

From the previously described robot head, the following requirements were set for designing the control electronics:

- At least eleven PWM signals for controlling the servos,
- At least nine digital outputs for controlling the LEDs,
- At least two free timers, besides the PWM generators,
- At least two USARTs.

For enhanced control of the LEDs, a timer is used to generate software driven PWM signal, so that various colors could be generated. The second timer is used as a system timer. The two USARTs are used for serial communication. One with a PC, that is running the face detection application, and the second one with other peripheral boards in the robot.

The Atmel Atmega2560 was chosen as the controller for the head's peripheral board. Out of the important characteristics for our design, it has sixteen PWM channels, seventy eight digital inputs/outputs, five timers and four USARTs. This controller exceeds our requirements, but is suitable for further development.

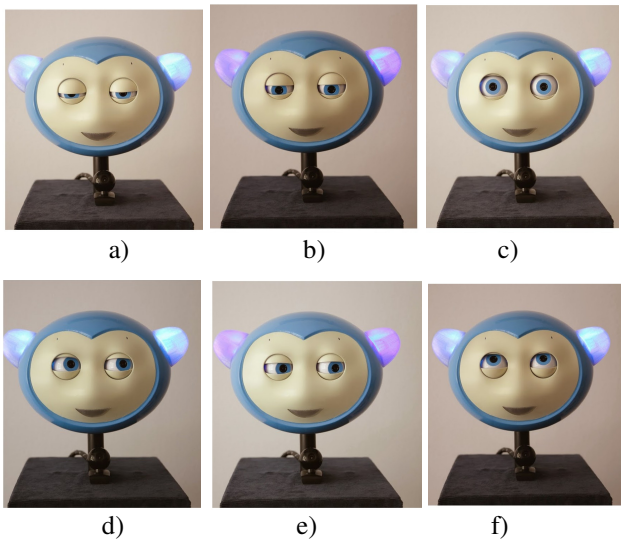


Figure 6. *The snapshots when MARKO is a) tired, b) sad, c) surprised, d) face tracking, e) looking left f) looking up*

The Atmega2560 and the servos are powered from a 5V rail. For generating the 5V, the LM2576 switch-mode regulator is used. This regulator can provide 3A of current, which is enough in steady state. Since the servos can draw a lot of current to overcome inertia, and there are eleven servos, three of these regulators are used. One regulator for each eye and its eyelids, and one for the eyebrows and the controller.

The behaviour of the robot is modeled as a finite state machine (FSM) with following states:

- sleeping;
- face tracking;
- facial expression;
- eye movement action.

Transition from one state to another is prescribed as follows. The initial state of the robot is "sleeping ". In that state robot is waiting for the verbal command from to user to wake up. Upon receiving the command to wake up, robot is transited into "face tracking" state. In this state robot is detecting the face of the nearest person standing in front of the robot. While in this state, for express the human-like behavior, the robot is blinking randomly in the time frame of 2-5 seconds. Depending on the voice commands perceived by dialog management system, the robot can transit from "face tracking" to any other state (Fig. 1.).

When the robot is in "facial expression" state it will show one of the following priory defined expressions:

- sad,
- tired and
- surprised.

When the MARKO is in "eye movement action" state it will show point its eyes up, down, left or right. If MARKO is in "facial expression" or "eye movement action" state, it will remain still for 3 second. After that the state is automatically changed back to " face tracking " and MARKO continues to track the person that is in front.

6 CONCLUSION

In this paper the concept and design of humanoid robot MARKO, an assistive technology for the cerebral palsy treatment, was presented. Short overview of the system modules including stereo vision system, speech recognition and synthesis system and dialog management system was given, as well as the mechanical design of the robot. Brief overview of the control architecture was presented, as well as the design of the control hardware platform for implementation of impedance control

Acknowledgments.

This work was funded by the Ministry of education and science of the Republic of Serbia under contract III44008 and by Provincial secretariat for science and technological development under contract 114-451-2116/2011. The authors would like to thank the company Dunkermotoren for its support and equipment donation.

References

[1] Iwata H. and Sugano S.: Design of Human Symbiotic robot TWENDY-ONE. In Proceedings of IEEE International Conference on Robotics and Automation, IEEE, Kobe, Japan, pp. 580-586, 12-17 May, (2009)

[2] Wade E., Parnandi A., Mead R. and Matarić M.: Socially Assistive Robotics for Guiding Motor Task Practice. PALADYN Journal of Behavioral Robotics, Vol. 2, No. 4, pp. 218-227, (2011)

[3] Malik N. A., Shamsuddin S., Yussof H., Miskam M. A. and Hamid A. C.: Feasibility of using a humanoid robot to elicit communicational response in children with mild autism. In Proceedings of IOP Conference Series: Materials Science and Engineering, Vol. 53, Conference 1, (2013)

[4] Ranatunga I., Rajruangrabin J., Popa D. O. and Makedon F.: Enhanced Therapeutic Interactivity using Social Robot Zeno. In Proceedings of PETRA '11, 4th International Conference on Pervasive Technologies Related to Assistive Environments, Crete, Greece, No. 57, May 25 - 27, (2011)

[5] Kozyavkin V., Kachmar O. and Ablikova I.: Humanoid Social Robots in the Rehabilitation of Children with Cerebral Palsy. Research workshop “ICTs for improving Patient Rehabilitation Research Techniques”, Oldenburg, Germany, May 20, (2014)

[6] Calderita L. V., Bustos P., Suárez Mejías C., Ferrer González B. and Bandera A.: Rehabilitation for children while playing with a robotic assistant in a serious game. In Proceeding of NEUROTECHNIX 2013, Vilamoura, Portugal, 18-20 September, (2013)

[7] Savić S., Raković M. and Borovac B.: Mechanical design and control algorithm for the arm of humanoid robot MARKO. In Proceedings of 6th PSU-UNS International Conference on Engineering and Technology ICET 2013, Novi Sad, Serbia, (2013)

[8] Tasevski T., Nikolić M. and Mišković D.: Integration of an Industrial Robot with the System for Image and Voice Recognition. Serbian Jour of Electrical Engineering, vol. 10, no. 1, pp. 219-230, (2013)

[9] Gnjatović M.: Therapist-Centered Design of a Robot’s Dialogue Behavior. Cognitive Computation, Springer US, Published online 13.05.2014, ISSN: 1866-9964, DOI: 10.1007/s12559-014-9272-1

Motion Primitives for Realization of Robust Humanoid Walk in Unstructured Envioronment

Raković Mirko, Borovac Branislav, Nikolić Milutin, Savić Srđan
Faculty of Technical Sciences
University of Novi Sad
Trg Dositeja Obradovića 6, 21000, Novi Sad, Serbia
e-mail: rakovicm@uns.ac.rs

ABSTRACT

Agile and robust walk of humanoid robots is a prerequisite for their broad applications in human's everyday life and their activity in the unstructured environment. In this paper a new framework for biped motion synthesis is described. The framework is based on use of simple movements (primitives) which are combined in order to obtain more complex movements. For realization of the on-line modifiable walk, relationship is established between the gait characteristics and the parameters of the primitives. The procedure for generating dynamically balanced humanoid robot walk is described in order to validate concept presented in framework.

1 INTRODUCTION

We can say that humans are well trained for locomotion because they are permanently practicing their movements from the very birth and over time they gain the experience in making decisions when and how the particular movement should be realised. The basic prerequisite for using biped robots in close environment with humans is ability to walk successfully. This imposes the requirement for online walk synthesis and modification on the basis of current situation.

The first method for the synthesis of dynamically balanced walking robot i.e. the semi-inverse method, was introduced by Juričić and Vukobratović [1]. In semi-inverse method, motion of one part of the system (feet) is prescribed, and the motion of the rest of the system is calculated so to ensure a dynamically balanced walk.

Synthesis of the walk for the majority of modern humanoid is based on prescribing the footprint position [2]. The basic idea in these approaches is to set the position and the time instances of the footprints on the ground. Based on that, the motion of legs can be determined, and the motion of the leg's joints can be calculated. To ensure the dynamically balanced walk, it is also necessary to predefine the reference zero moment point (ZMP) trajectory, which is used to calculate the motion of the rest of the system, i.e. the trunk and arms.

However, for the use of humanoids in human's environment, the realization of walk can't be programmed in advance but have to be synthesized on-line. Since the mechanical structure of the humanoid robot is similar to those of human, the biologically inspired approach is more promising for realization of biped walking. Neurological studies of walk proved the existence of the patterns of leg motion which are constantly re-executed and over time upgraded and refined [3-5]. Studies of walk also have shown that there are similar short movements that are repeated in different types of walk (for instance, during regular walk and climbing stairs).

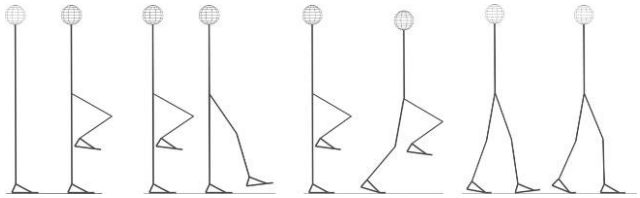
2 MOTION PRIMITIVES

The application of primitives for the generation of the movements is not an entirely new approach, but different authors define primitives in different ways. In [6], the author introduced the notion of Dynamic Movement Primitives (DMP), defined as the desired state of the kinematics of the extremities, obtained by prescribing in advance the values of the angles, angular velocities and angular accelerations for each robot joint. The work [7] gives a library in which each primitive represents one step. Based on the preset requirements and the current state of the robot, a new step which corresponds to the requirements is selected from the library. In [8], the authors used the leg motion primitives that are obtained by segmenting the movements recorded from man. The recorded primitives are modified so to adjust to the kinematic and dynamic parameters of the robot, to preserve the form of the recorded motion, while maintaining its dynamic balance. In the base of primitives described in [9] each primitive satisfies the conditions of periodicity. Thus the primitives from the base allow the changes in the motion direction, adjustment of the step length, and passing over the obstacles.

The approach presented here differs in several important points from those mentioned above. We define primitive as a simple movement which is realized by simultaneous and synchronized motion of a number of joints [10]. Illustrative examples are presented in Fig. 1. The complex motion (i.e. walking) is composed by combining and tying different primitives, without having a reference trajectory that was prescribed in advance.

Each primitive is parameterized, so that a form of the realization is adjusted by choosing the appropriate parameters, even during execution. There are no limitations in the defining of new primitives, either in the number of

parameters or in the number of joints that will be activated. Every primitive can be considered as a movement to be realized on the basis of the task assigned and the current situation. To define execution trajectory, we propose to associate coordinate frame to a particular link of the kinematic chain performing the primitive and then to assign the target positions to which is necessary to bring it by the realization of the selected primitive. Another important point is smooth continuation of tied primitives.



a) b) c) d)

Figure 1. Stick diagrams of the primitives: a) leg bending, b) leg stretching, c) inclining the robot forward, d) making the foot surface contact

For the gait synthesis, we introduced five primitives realized by the legs, along with one realized by the trunk and one by the arms. The primitives that are realized by the legs in the single-support phase are: bending of the leg in the swing phase, stretching of the leg in the swing phase, and inclining the robot forward. The primitives realized during the double-support phase are: making the foot surface contact after the heel strike, and transferring the body weight onto the subsequent supporting leg. The primitives that are realized by the trunk and arms are: maintenance of the trunk upright posture, and arms swinging during the walk.

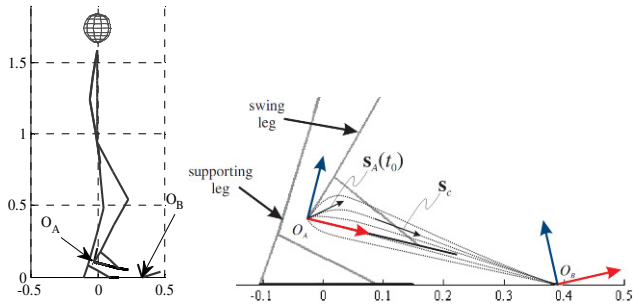
The trunk is used to keep the upright posture and at the same time to maintain dynamic balance. However, the maintenance of dynamic balance is of higher priority, and task assigned to trunk at each time instant depend on ZMP position whose desired position is predefined for each primitive. If actual ZMP distance from its desired position exceeds threshold, it has been considered dynamic balance is jeopardized, and the trunk is used to return the ZMP back, to prevent the system from falling down. When dynamic balance is not jeopardized, it is possible to tackle the task of correcting deviations of the trunk from its upright position.

3. WALK SYNTHESIS

3.1 Primitives smooth tying

Let us assume that the locomotion mechanism moves and that the primitive realized by the leg in swing phase has just been completed, so that at the beginning of execution of the next primitive the leg is not at rest. While walking, the robot finds itself in such a situation when, in the single-support phase (Fig. 2 a)), the leg bending is ended, and the realization of leg stretching is about to start. The vector $\mathbf{r}_A$ represents the instantaneous position of the coordinate frame O_A (in Fig. 2 a) it is the heel tip), while the vector $\mathbf{r}_B$ represents the target position (defined by the coordinate frame O_B) to which O_A is to be brought. In view of the fact that the swing leg in the given moment is moving, it is necessary to determine first the desired velocity at point A. i.e. $\mathbf{s}_A = [\mathbf{v}_A \quad \boldsymbol{\omega}_A]^T$ of the coordinate frame O_A to ensure that the leg stretching follows smoothly after the just completed leg bending.

The trajectory (as well as the velocity $\mathbf{s}_A$ in each moment) by which the foot is to move from the starting position O_A to the target position O_B depends on the intensity and direction of the velocity $\mathbf{s}_A^0$ at the starting moment t_{p0}. Fig. 2 b) shows a set of possible paths from O_A to O_B.



a) b)

Figure 2. Stick diagram of the robot for leg stretching: a) robot posture with the preset target position OB to which the coordinate frame on the heel tip OA should be brought b) possible paths of OA for different initial velocities.

The desired velocity $\mathbf{s}_A$ is calculated as:

$$\mathbf{s}_A(t_i) = (1\text{-}b(t_i)) \cdot \mathbf{s}_A^0 + b(t_i) \cdot \left[v_{int} \cdot \mathbf{p}_e^{ort} \quad \omega_{int} \cdot \mathbf{o}_e^{ort} \right]^T$$

where $\mathbf{p}_e^{ort}$ and $\mathbf{o}_e^{ort}$ represent the orts of the $\mathbf{p}_e = \mathbf{p}_B - \mathbf{p}_A$ and $\mathbf{o}_e = \mathbf{o}_B - \mathbf{o}_A$ i.e. the position and orientation between $\mathbf{r}_A$ and $\mathbf{r}_B$. The coefficient b changes during the prescribed time interval from 0 to 1. This ensures smooth tying of primitives by a gradual change of the velocity $\mathbf{s}_A(t)$ from the initial value $\mathbf{s}_A^0$ to the value that will lead the frame O_A to the target position.

Intensities of the linear and angular velocities v_{int} and ω_{int} are dependent on the cruising speeds v_c and ω_c, which are set by the primitive parameters. To ensure the leg gradual stopping the intensities of the velocities v_{int} and ω_{int} are to be reduced when O_A comes close to the target. Having thus determined $\mathbf{s}_A(t_i)$, and using inverse kinematics, the desired joint angular velocities can be calculated.

3.2 Joints motion control and preservation of dynamic balance

For control synthesis we propose cascade control loop as shown in Fig. 3. The inputs to the first block are parameters

of the desired overall motion. In this block, the primitives are combined by calling the functions for each primitive used. To check the fulfillment of the conditions for starting and ending the realization of each primitive the inputs to this block are also the feedback values of the robot instantaneous state (angles and angular velocities, along with the current positions of ZMP and vertical projection of the mass centre-PCM). The outputs from the first block are the desired values of joint angular velocities ${}^{des}\dot{\mathbf{q}}(t_i)$ as well as the desired positions of ZMP and PCM. The role of the second block is to ensure the maintenance of dynamic balance which is performed by corrections of the desired angular velocities ${}^{des}\dot{\mathbf{q}}(t_i)$.

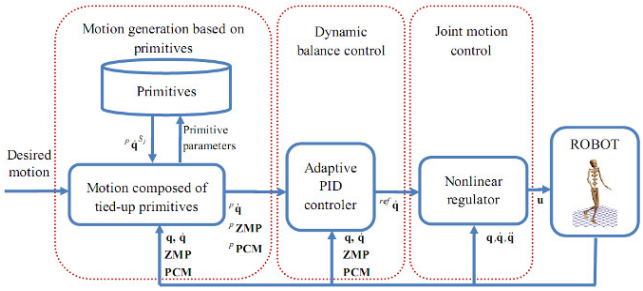


Figure 3. Block diagram of the robot control for realization of the motion synthesized

Corrections are determined on the basis of the desired and the current values of ZMP and PCM, as well as of the current values of the joint angles and angular velocities $\mathbf{q}$ and $\dot{\mathbf{q}}$ [11]. The corrections of the angular velocities depend on the deviations of ZMP and PCM from the desired values that are generated online with respect to current position of feet. In [12] was shown that the movements of the ankle joint in one direction and of the hip in the other direction ensured a very efficient control of the positions of PCM and ZMP. An different approach can be seen at [13].

After determining the reference angular joint velocities ${}^{ref}\dot{\mathbf{q}}(t_i)$ the voltages of the motors have to be calculated. For realization of this motion we used nonlinear regulator which is a combination of feedback linearization, sliding mode control and disturbance estimator [14].

3 SIMULATION RESULTS

As a overall walk parameters we introduced: walk speed W_S, height to which the foot is lifted during the swing phase F_H, step length S_L, and walking direction W_D. All these parameters can be changed within an predefined range, but the basic walk with the parameters: W_S=1, F_H=1, S_L=1, W_D=0 should be synthesized first.

The gait speed (W_S) has a direct influence on the speed of execution of each primitive, F_H influences the height to which the foot is to be lifted during leg bending, the step length (S_L) influences the parameters for leg stretching. The desired gait direction W_D, (the turn angle with respect to the x axis of the fixed coordinate frame) influences also the parameters of the primitive for leg stretching.

The synthesis of one half-step is performed in four phases involving the execution of the corresponding primitives that are realized by the legs. Also, for each phase is given the desired goal position of the PCM. Simultaneously with the primitives that are realized by the legs, irrespective of the current phase, the primitive for keeping the trunk in an upright position and the primitive for arms swinging are constantly executed during the walk.

Fig. 4 shows the stick diagrams of the robot, disposition of the feet, and positions of the ZMP and PCM during the realization of the basic motion by the previously described procedure. The trajectories of ZMP and PCM were all the time inside the support area, i.e. the robot's dynamic balance was constantly preserved.

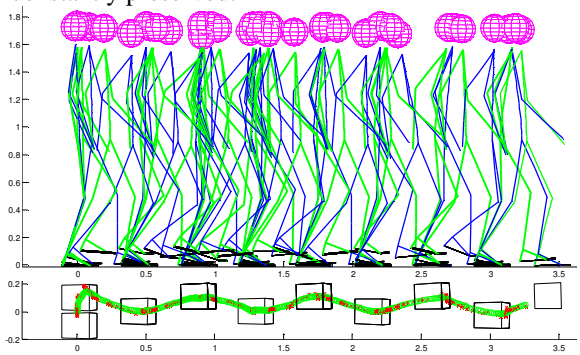


Figure 4. Stick diagram of the robot, positions of the feet, and the trajectories of ZMP and PCM for the basic walk

Fig. 5 presents the case when the robot is online adjusting the pace, direction of the walk and height of the foot during the swing phase in order to pass between the tables and step over the bar on the ground. It can be seen that walk combined from primitives is online modified by changing the overall parameters of walk.

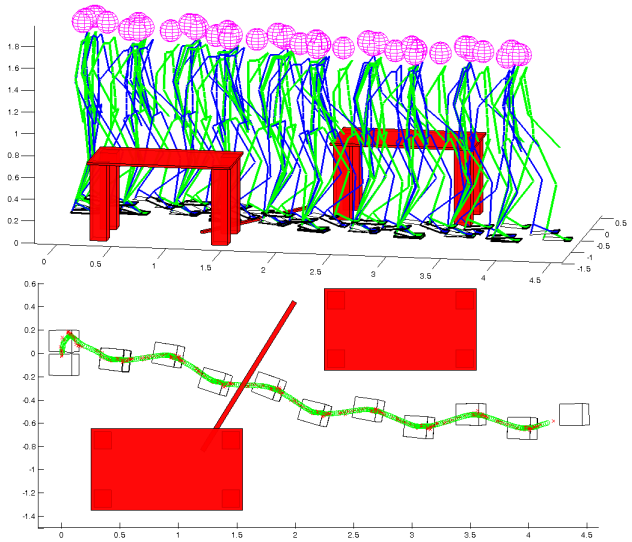


Fig. 5. Example of on-line change of the walking direction, foot height during the swing phase and speed to pass around the obstacles

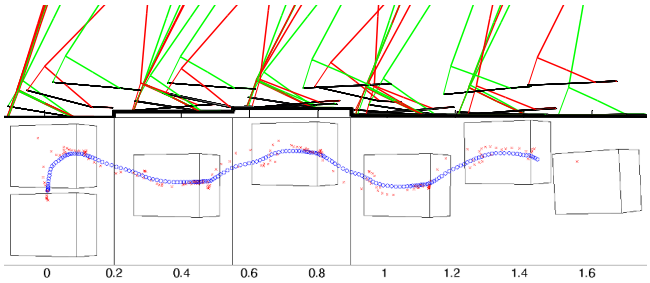


Fig. 6. Stick diagram of the robots' legs, positions of the feet, and the trajectories of ZMP (crosses) and PCM (circles) for t he case when robot is walking on uneven terrain with unknown terrain configuration

In Fig. 6. is shown the walk of a robot over the irregular terrain. The terrain is uneven with different heights of the ground surface. The starting and ending conditions for primitives execution made possible for robot to adapt to such a terrain. At the end of first half-step ground is 2cm above the initial ground level, at the end of second half-step it is 3cm above the ground level, and finally, during third step it is 1cm above the ground level. The robot performed walk without knowing in advance the configuration of the terrain.

4. CONCLUSION

In this work we proposed use of primitive based bipedal walk for motion in unstructured and irregular environment. The walk was composed of tied primitives without exact reference to be followed. Motion was generated online and modified according to current situation. The preservation of dynamic balance was also secured by monitoring the deviation of the actual ZMP from its desired position.

The main achievement of this paper is that has been shown that walk combined from primitives can easily be performed and modified online by simply changing parameters of the overall motion (turn left or right, step over obstacle, …). This was achieved by establishing the relationship between the primitive parameters and the parameters of the overall motion which can be estimated, for example, from visual information (robot can see obstacle in front of it and estimate its size and position). The presented approach is also robust and adaptable to unknown terrain configuration because primitive will be executed only if its execution is feasible i.e. if conditions for primitive execution exist. This enables versatile and robust walk.

Acknowledgments.

This work was funded by the Ministry of education and science of the Republic of Serbia under contract III44008 and by Provincial secretariat for science and technological development under contract 114-451-2116/2011.

References

[1] Juričić D., Vukobratović M.: "Mathematical modeling of a bipedal walking system" *In Proceedings of ASME Winter Annual Meeting*, pp. 26–30, 1972

[2] Morisawa M., Harada K., Kajita S., Nakaoka S., Fujiwara K., Kanehiro F., Kaneko K., Hirukawa H.: "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," *In Robotics and Automation, 2007 IEEE International Conference on. IEEE*, pp. 3989–3994, April 2007.

[3] Breniere Y., Bril B.: "Development of postural control of gravity forces in children during the first 5 years of walking", *Experimental Brain Research*, vol. 121, no. 3, pp. 255-262., 1998

[4] Evans S., Keenan T.: "*An introduction to child development*", Sage Publications Ltd., 2009.

[5] Mussa-Ivaldi F. A., Giszter S. F., Bizzi, E.: Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences*, *91*(16), 7534-7538, 1994

[6] Schaal S.: "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics", *In Adaptive Motion of Animals and Machines*, pp. 261-280. Springer Tokyo, 2006

[7] Hauser K., Bretl T., Harada K., Latombe, J. C.: "Using motion primitives in probabilistic sample-based planning for humanoid robots", *Algorithmic Foundation of Robotics VII*, pp. 507-522, 2008

[8] Zhang L., Bi S., Liu D.: "Dynamic Leg Motion Generation of Humanoid Robot Based on Human Motion Capture", *Intelligent Robotics and Applications,* pp. 83-92, 2008

[9] Denk J., Schmidt G.: "Synthesis of walking primitive databases for biped robots in 3D-environments", *In Proceedings of IEEE International Conference on Robotics and Automation, ICRA'03*, pp. 1343-1349, 2003

[10] Raković M., Borovac B., Nikolic M., Savic S.: "Realization of Biped Walking in Unstructured Environment Using Motion Primitives", *Robotics, IEEE Transactions on*, Vol. 30, No.6, in press, 2014

[11] Raković M., Borovac B., Nikolic M., Savic S.: "Biped Walking on Irregular Terrain Using Motion Primitives", *Proceedings of Romansy 2014 XX CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Vol. 22, pp 265-273, 2014

[12] Vukobratović M., Herr H., Borovac B., Raković M., Popovic M., Hoffmann A., Jovanović M., Potkonjak V.: "Biological principles of control selection for a humanoid robot's dynamic balance preservation." *International Journal of Humanoid Robotics*, Vol. 05, No. 04, pp. 639–678, 2008.

[13] Peuker F., Maufroy C., Seyfarth A.: "Leg-adjustment strategies for stable running in three dimensions", *Bioinspiration and Biomimetics*, Vol. 7, no. 3, 2012.

[14] Slotine J. J. E., Li W., et al.: *"Applied nonlinear control"*. Prentice hall New Jersey, 1991,

FORCE BASED SURFACE FOLLOWING BASED ON ITERATIVE LEARNING

Bojan Nemec, Tadej Petrič, Aleš Ude
Department of Automatics, Bicybernetics and Robotics
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773565; fax: +386 1 4251038
e-mail: bojan,nemec@ijs.si

ABSTRACT

The paper addresses a problem of following a previously unknown surface. Such unknown surfaces may appear in many industrial applications as well as in field of home robotics. By applying Iterative Learning Framework (ILC) the robot is able to efficiently learn the shape of an unknown surface by itself in only a few iterations. In this paper we extended the Standard ILC approach by incorporating the idea of on-line coaching, which additionally improved the learning speed and robustness.

1 INTRODUCTION

In many industrial applications the robot is required to follow a previously unknown surface. Typical operations which involve this problem are polishing, grinding, cleaning, etc. [1]. This problem is relevant also for the future generation of home robots while performing task from everyday life. Such tasks are for example cleaning the table, polishing furniture, etc. [2]. To solve this problem the force control [3, 4] can be applied. Clearly, if the robot is able to move around while precisely maintaining the contact force, the surface shape can be directly captured from the robot's motion. Force control algorithms can be portioned into the admittance force control and torque based control [5]. For example, admittance control tracks the desired force by controlling the robot end effector displacement and is as such sensible to the changes in the environment stiffness. Torque based control has no such limitations and generally results in better, faster and more stable force tracking [5]. On the other hand, even well designed force controller which decouples and compensates the non-linear robot dynamics exhibits a lag in following inclined surface, which may results in a large force tracking error. This problem can be efficiently solved applying Iterative Learning Control paradigm (ILC) [6, 7]. In industry as well as in natural home environments there are many tasks that need to be repeated on and on. In such a case, humans can acquire skills by repeating and improving the same action all over again. The same principle can be adopted also in machine motor control theory, when a system follows a similar trajectory repeatedly. In this case, the information about the tracking error can be used to improve performance in the next repetition of the same trajectory. This is the main idea of the iterative learning control.

In the paper, we propose a method for force tracking based on Iterative Learning Control. Main advantage of the proposed method is that it can be successfully implemented also in aggregation with the admittance force control, which significantly simplifies the implementation [8,9]. We also demonstrated experimentally that the proposed control can cope with variable environmental stiffness. In our approach, we applied Dynamic Motion Primitives (DMP) framework [10] for the underlying representation of the position/orientation and force/torque trajectories. As the extension of the ILC framework, we further propose another approach which combines the ideas of ILC and on-line coaching [11].

2 TRAJECORY PARAMETRISATION WITH DYNAMIC MOTION PRIMITIVES

Our approach relies on parameterized trajectories using Dynamic Movement Primitives [10]. Within this framework, motion trajectories for every degree of freedom are described by its own dynamic system, but with a common phase to synchronize them. For point-to-point movements, given either in joint or in task space, the trajectory *y* of each robot degree of freedom is described by the following system of nonlinear differential equations

$$\begin{aligned} \tau \dot{z} &= \alpha_z(\beta_z(g-y)-z)+f(x), \\ \tau \dot{y} &= z, \\ \tau \dot{x} &= -\alpha_x x, \end{aligned} \tag{1}$$

where x is the phase variable, z is an auxiliary variable describing scaled velocity, τ is the time variable and α_z, β_z and α_x are gains chosen in such a way that the system is critically damped and converges to the unique equilibrium point $(z, y, x) = (0, g, 0)$. The nonlinear term modifies the dynamics of the second-order system in order to approximate any smooth trajectory from the initial

position y_0 to the final position g. It is formed of weighted sum of n radial basis functions,

$$f(x) = \frac{\sum_{i=1}^{n} w_i \psi_i(x)}{\sum_{i=1}^{n} \psi_i(x)}, \quad \psi_i(x) = \exp(-h_i(x-c_i)^2), \quad (2)$$

where c_i and h_i are the centers of the radial basis functions and their widths, respectively. Weights w_i are calculated using the regression in such a way, that the DMP trajectory y matches the desired trajectory y_d [12]. Using this framework, we encode the initial robot trajectory for each degree of freedom. The benefits of using DMP framework for trajectory representation are numerous; trajectories can be modulated, time scaled, stopped, slowed down, goal change will preserve smooth trajectory, etc. Signals like forces and torques having no goal position are usually represented only with radial basis functions (2).

2 SURFACE FOLLOWING WITH ADDMTTANCE ILC

In this section we will present ILC for the hybrid admittance force control. In hybrid force control [13], each degree of freedom in operational space is controlled either by position/orientation or force/torque, respectively. The position vector feed to the position controller $\mathbf{y}_c$ is calculated according to

$$\begin{aligned} \mathbf{y}_c(t) &= \mathbf{S}_p \mathbf{y}_d(t) + \mathbf{S}_f(\mathbf{K}_{fp}\mathbf{e}_f(t) + \mathbf{K}_{fi}\int \mathbf{e}_f(t)dt), \\ \mathbf{e}_f(t) &= \mathbf{R}(\mathbf{F}_d(t) - \mathbf{F}_m(t)), \end{aligned} \quad (3)$$

where t denotes time variable, $\mathbf{S}_p$ and $\mathbf{S}_f$ are diagonal and orthogonal position and force selection matrices, $\mathbf{y}_d$ is the desired position/orientation vector, $\mathbf{F}_d$ and $\mathbf{F}_m$ are the desired and measured end effector forces/torques in the tool coordinates, $\mathbf{R}$ is the robot tool center point rotation matrix and $\mathbf{K}_{fp}, \mathbf{K}_{fi}$ are appropriately chosen positive definite force control gains. It can be easily verified, that the controller (3) is not capable of tracking constant environment slope with zero force error. To overcome this limitation, we apply ILC, which learns feed-forward term φ_l in several repetitions (cycles) of the same task. The time discrete form of ILC is

$$\begin{aligned} \mathbf{y}_{c,l}(k) &= \mathbf{S}_p \mathbf{y}_{d,l}(k) + \mathbf{S}_f(\mathbf{K}_{fp}\mathbf{e}_{f,l}(k) + \mathbf{K}_{fi}\sum_{i=0}^{k}\mathbf{e}_{f,l}(i) + \varphi_l(k)), \\ \varphi_l(k) &= \mathbf{y}_{c,l-1}(k) + \mathbf{K}_{fl}\mathbf{e}_{f,l-1}(k+d) \end{aligned} \quad (4)$$

where $\mathbf{K}_{fl}$ is ILC gain and subscript l denotes $l-th$ cycle of the same task. Note that the error signal from the previous cycle is shifted for d samples in order to cope with the delays incorporated in the robot control system. Standard ILC assumptions include: 1) Stable system dynamics, 2) System returns to the same initial conditions at the start of each trial, 3) Each trial has the same length. In the first repetition, error signals $\mathbf{e}_{f,l-1}$ and $\mathbf{y}_{c,l-1}$ are set to zero. Thus, in the first trial φ_l equals to zero and (4) is identical to the classical admittance control given by (3).

3 SURFACE FOLLOWING WITH RECURSIVE REGRESSION ILC (RRILC)

In this section we propose an improvement of the ILC, where the feed-forward term $\varphi_l(k)$ is a nonlinear function represented with m radial basis functions similar as in (2).

$$\varphi_l(x(k)) = \frac{\sum_{i=1}^{m} w_i \psi_i(x(k+d))}{\sum_{i=1}^{m} \psi_i(x(k+d))}, \quad (5)$$

Note that this representation is phase dependent. Therefore the corresponding phase variable $x(k)$ needs to be calculated at every discrete time sample k. As in the previous section, parameter d compensates for the delays incorporated in the robot control system The corresponding weights $\mathbf{w}_j$ for each degree of freedom j are calculated recursively using the error signal $e_j(k)$ according to the

$$\begin{aligned} \mathbf{w}_j(k) &= \mathbf{w}_j(k-1) + \left(K_r e_j(k) - \mathbf{a}_j(k)\mathbf{w}_j(k-1)\right)\mathbf{P}_j(k)\mathbf{a}_j(k), \\ \mathbf{P}_j(k) &= \mathbf{P}_j(k-1) + \frac{\mathbf{P}_j(k-1)\mathbf{a}_j(k)\mathbf{a}_j^T(k)\mathbf{P}_j(k-1)}{1+\mathbf{a}_j^T(k)\mathbf{P}_j(k-1)\mathbf{a}_j(k)}, \\ \mathbf{a}_j(k) &= \left[\frac{\psi_1(x(k))x(k)}{\sum_{i=1}^{n}\psi_j(x(k))} \quad \dots \quad \frac{\psi_n(x(k))x(k)}{\sum_{i=1}^{n}\psi_j(x(k))}\right]. \end{aligned} \quad (6)$$

K_r is the gain of the recursive regression ILC. This new formulation of ILC generates feed-forward signal φ_l from the error signal taken in the current iteration e_j and using the Gaussian kernel functions as a statistical model. On contrary, standard ILC uses signals from the previous iteration and therefore has to complete entire iteration before generating new feed forward signal.

4 SIMULATION RESULTS

In this section we will compare the results of both proposed algorithms obtained in simulated environment. The simulated work cell consisted of two KUKA LWR 7 DOFs robot equipped with Barrett hands. One robot was holding a paint roller in the hand. The task was to follow the previously unknown surface while maintaining the constant 5 N force in Z direction with constant speed 0.15 m/s in Y direction. The shape of the object was double triangle as illustrated in fig 1. Simulation was accomplished in Matlab, where we implemented control law and performed simulation of the robot and environment dynamics. The initial position/orientation trajectory $\mathbf{y}_d$ was encoded as DMP using (1). The desired force profile was encoded with radial basis functions (2). The simulated environment stiffness was 1000 N/m and the controller gains were set to $\mathbf{K}_{fi} = \mathbf{I}\, 5 \cdot 10^{-4}$, $\mathbf{K}_{fp} = \mathbf{I}\, 0.01$, $\mathbf{K}_{fl} = \mathbf{K}_{fp}$ and $K_r = 4 \cdot 10^{-4}$.

Best results were obtained with the sample shift $d = 3$. The sampling frequency of the control loop was 100 Hz. The robot and environment dynamics were updated at 1 KHz rate. 5 learning cycles were performed.

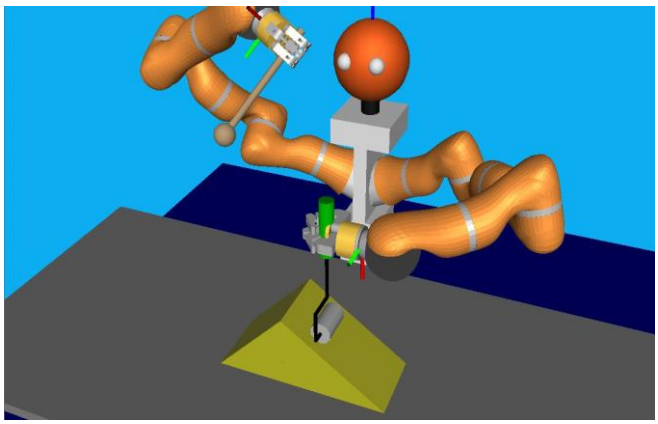


Figure 1: *Simulated setup.*

Figure 2 shows the simulated force and captured position in Z direction, respectively. For sake of clearness only results of the 1st and the 5th repetition are shown. As we can see, RRILC updates the feed forward term also in the 1st cycle and therefore updates faster during the learning. Another benefit of RRILC is more smooth response. Namely, smoothing of the feed forward compensation term φ_l is inherited from the representation with radial basis functions.

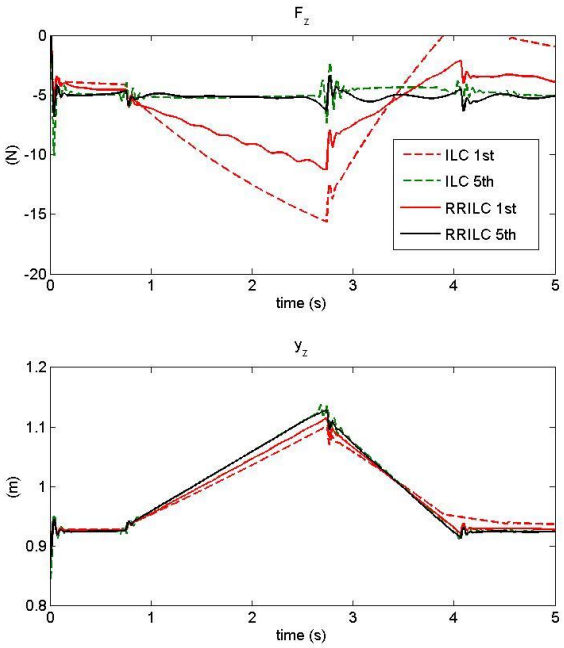


Figure 2: *Comparison of the simulation results obtained with ILC and RRILC. Upper graph shows the force tracking in Z direction. Lower graph shows Z coordinate of the captured surface.*

5 EXPERIMENTAL RESULTS

Both proposed algorithms were evaluated also experimentally. As in the simulation, the work cell consisted of two KUKA LWR 7 DOFs robot equipped with Barrett hands. The task was identical as in the simulated environment except that the speed in Y direction was set to 0.1 m/s and that the shape of the object was as in the Fig. 3. It was made of thin aluminum plate, on part of the plate was firmly attached to the bottom, while the other was not. Therefore, during the surface following the environment stiffness was changing from very stiff at the beginning to the very compliant at the end. The programmed robot stiffness was 1000 N/m and the controller gains were set to $\mathbf{K}_{fi} = \mathbf{I}\, 2 \cdot 10^{-5}$, $\mathbf{K}_{fp} = \mathbf{I}\, 0.002$, $\mathbf{K}_{fl} = \mathbf{K}_{fp}$ and $K_r = 2 \cdot 10^{-3}$.

Best results were obtained with the sample shift $d = 4$.

Figure 4 shows the experimentally obtained force and captured position in Z direction, respectively. For sake of clearness only results of the 1st and the 5th repetition are shown. Similar as in the simulation, RRILC updates the feed forward term also in the 1st cycle and therefore updates faster during the learning. Beside smoother response, RRILC exhibits less oscillation at the impact.

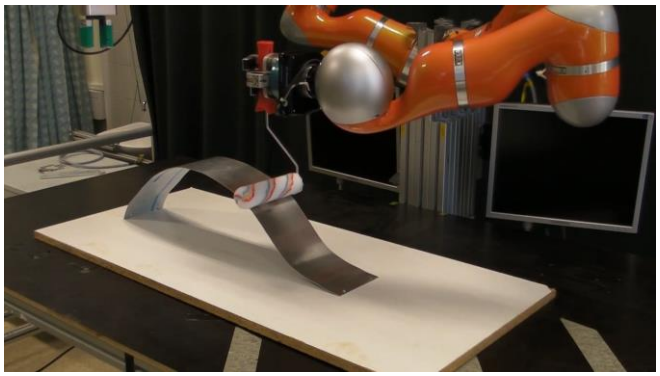


Figure 3: *Experimental setup.*

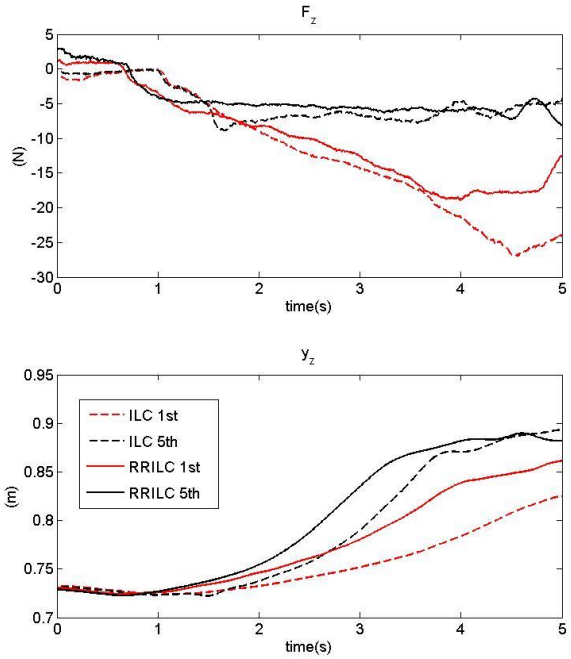


Figure 4: *Comparison of the experimental results obtained with ILC and RRILC. Upper graph shows the force tracking in Z direction. Lower graph shows Z coordinate of the captured surface.*

6 CONCLUSIONS

In the paper we proposed two approaches to learn the previously unknown shape of an object using tactile sensing. The shape was learned in a few iterations where the control goal was to maintain the desired contact forces while exploring the object shape. The first approach exploits the ILC framework in conjunction with the hybrid PI force control. In the second approach we propose new formulation of ILC, which learns compensation term using the current force tracking error and recursive regression for determining the weights of radial basis functions. Both approaches were evaluated in simulation and experimentally. Final results of both approaches show that newly proposed RRILC exhibits fewer oscillations and learns faster.

References

[1] R. Bogue. Finishing robots: a review of technologies and applications, Industrial Robot: An International Journal, Vol. 36, no. 1, pp. 6 – 12, 2009.

[2] C. C. Kemp, A. Edsinger, E. Torres-Jara, Challenges for Robot Manipulation in Human Environments, IEEE Robotics & Automation Magazine, vol. 14, no. 1,pp 20–29, 2007.

[3] H. Bruyninckx, J. De Schutter. Specification of Force-Controlled Actions in the "Task Frame Formalism"-A Synthesis, IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp 581–589, 1996.

[4] T. Tsujimura, T. Yabuta. Object detection by tactile sensing method employing force/torque information, Robotics and Automation, IEEE Transactions on , vol.5, no.4, pp.444–450, 1989.

[5] L. Villani and J. De Schutter. Force control, in Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, pp. 161–185, 2008.

[6] D. Bristow, M. Tharayil, and A. Alleyne. A survey of iterative learning control, Control Systems, IEEE, vol. 26, no. 3, pp. 96 –114, 2006.

[7] K. L. Moore, Y. Chen, and H.-S. Ahn. Iterative learning control: A tutorial and big picture view, in IEEE Conference on Decision and Control, San Diego, CA, pp. 2352 –2357, 2006.

[8] A. Gams, B. Nemec, A. Ijspeert, and A. Ude. Coupling movement primitives: Interaction with the environment and bimanual tasks, IEEE Transactions on Robotics, vol. 30, no. 4, pp. 816–830, 2014

[9] B. Nemec, F. Abu-Dakka, J. A. Jørgensen, T. R. Savarimuthu, B. Ridge, J. Jouffroy, H. G. Petersen, N. Krueger, and A. Ude. Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile, in International Conference on Advanced Robotics, Montevideo, Uruguay, 2013.

[10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors, Neural Computation, vol. 25, no. 2, pp. 328–373, 2013.

[11] T. Petrič, A. Gams, L. Žlajpah, A. Ude, and J. Morimoto. Online approach for altering robot behaviors based on human in the loop coaching gestures, in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014.

[12] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives, IEEE Transactions on Robotics, vol. 26, no. 5, pp. 800–815, 2010.

[13] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulator. ASME Journal of Dynamic Systems, Measurement and Control, vol. 103, pp. 126–133, 1981.

CONFIGURATION BASED COMPLINANCE CONTROL OF INDUSTRIAL HUMANOIDS

Petar B. Petrović, Nikola Lukić, Ivan Danilov
CyberManufacturing Systems Laboratory
Faculty of Mechanical Engineering, University of Belgrade
Kraljice Marije 16, 11120 Belgrade, Serbia
Tel: +381 11 3302435
e-mail: pbpetrovic@mas.bg.ac.rs

ABSTRACT

This paper proposes the Configuration-based Stiffness Control (CSC) method for kinetostatically consistent control of robot compliant behavior, based on the gradient projection of the cost function which minimizes the norm of off-diagonal elements of the jointspace matrix. Validity of the proposed method is tested by simulation experiments.

1 INTRODUCTION

Regardless of mechanical and kinematical composition, any robot is an elastomechanical structure which deforms while interacting with the environment. Contrary to other elastomechanical mechanisms, flexibility of robot mechanism and its compliant behavior can be actively controlled.

Compliance control is a classical research topic, studied for decades by many researchers, [1, 2, and 3]. However, due to mechanical, sensing and control complexity, compliance control is still an open and very challenging research topic. Recent progress in the development of soft and humanoid robots, especially their extensive implementation in complex tasks with intensive human-robot physical interaction as well as in constrained motion tasks where precise morphing of the robot taskspace stiffness is essential for successful task execution, increase the importance of compliance research, [4].

The robot compliant behavior, which is considered in this report, is limited to the kinematically redundant robot mechanisms, driven by monoarticular variable stiffness actuators. Despite of its computational complexity, kinematic redundancy is a source of freedom in task execution [6], because it provides the robot mechanism with an increased level of dexterity. Kinetostatical consistency is provided by gradient projection of the joint displacement vector generated by an appropriate cost faction onto the Jacobian null space. Based on kinematic redundancy the Configuration-based Stiffness Control (CSC) method is formulated as a kinetostatically consistent framework for robot compliant behavior control.

2 JOINTSPACE STIFFNESS MATRIX

Considering the kinetostatic domain, serial linkage of the robot arm mechanism and its actuation system can be reduced to a generalized nonlinear spring as shown in Figure 1.

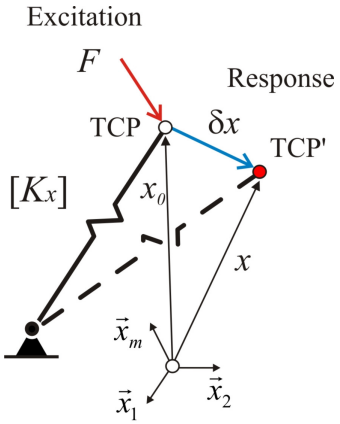


Figure 1: *Serial robot linkage reduced to a generalized Cartesian spring.*

Nonlinear behaviour of this elastomechanical structure can be locally linearized, which leads to the following relation between the external excitation force vector $F \in R^m$ (robot-environment interaction force) and the corresponding response, i.e., the displacement vector $\delta x \in R^m$ of the robot tool center point (TCP):

$$F = K_x \delta x \tag{1}$$

Linear transformation operator between two vector spaces (1) is the generalized stiffness matrix $K_x \in R^{m \times m}$, which is expressed in the robot taskspace Cartesian coordinates.

Mapping of external force from the robot taskspace to the corresponding joint torques vector, and vice versa can be derived by the energy conservation principle [5], which leads to:

$$\tau = J^T(q) F \tag{2}$$

By definition, joint-space stiffness matrix can be directly derived from (2) and (1):

$$K_q = -\frac{\partial \tau}{\partial q} = J^T(q) K_x J(q) - [\frac{\partial}{\partial q} J^T(q)] K_x \delta x \quad (3)$$

where $J(q) \in R^{m \times n}$ denotes Jacobian matrix. The second term in (3) is usually neglected since the displacement vector δx is very small when the robot arm TCP is close to its equilibrium configuration. This leads to the relation which is well known from the early stages of research in the field of active stiffness control, [3]:

$$K_x \rightarrow K_q(q, K_x) = J^T(q) K_x J(q), \; K_q \in R^{n \times n} \quad (4)$$

However, the adopted approximation generates significant integration errors when some external force F is present at the robot TCP [14, 15]. In this case the congruence transformation (4) becomes physically inconsistent in terms of its conservativeness properties, even though the symmetry of the joint-space stiffness matrix is preserved. Therefore, the mapping (3) can be understood as a Conservative Congruence Transformation, [6].

The joint-space stiffness matrix resulting from (3) and (4) is generally nondiagonal. Today, robots are almost exclusively driven by monoarticular actuators. Regardless of their mechanical design and the control concept applied, off-diagonal nonzero elements are impossible to be physically generated in such robots. Of-diagonal elements require polyarticular actuators, i.e., actuation redundancy. But, this kind of actuation is technically very complex to be practically realised. In order to solve this problem, an analytical framework which is based on kinematic redundancy has been developed.

3 NULLSPACE STIFFNESS CONTROL

Within the nullspace N($J(q)$) of the kinematically redundant robot with $r = (n - m)$ redundant degrees of freedom ($r \geq 1$), exists at least one configuration subspace $Q \subseteq$ N($J(q)$) with a nonempty set of configurations (robot postures) $q^* \in Q \wedge Q \neq \varnothing$ which simultaneously satisfies following constraints:

c1: The vector q^* is consistent with the desired position x_d of the robot TCP point – the primary task constraint,

c2: The vector q^* is consistent with the desired generalized stiffness K_{xd} of the robot TCP - the secondary task constraint, and

c3: The vector q^* is consistent with the canonical form of the corresponding actuation stiffness matrix K_{qd} such that K_{qd} is the congruence transformation (7) or (8) of the desired stiffness matrix of the robot TCP.

The above given proposition is based on the assumption that the increased mobility/dexterity of the kinematically redundant robot mechanism, which is actuated by a set of variable stiffness monoarticular actuators, is sufficient to effectively induce desired, or sufficiently good approximation of the desired generalized stiffness matrix of the robot TCP.

3.1 Nullspace complementary projector

This leads to the definition of the generalized virtual displacement which can be analytically expressed by the following relation:

$$\delta q = \delta q_P + \delta q_N = J^+(q) \delta x + P^c_{N(J)}(q) \delta q_0 \quad (5)$$

where: $J^+(q)$ is the generalized inverse of the Jacobian matrix $J(q)$ which satisfies Moore-Penrose least norm condition, δq_0 is an arbitrary vector from the robot configuration space, and $P^c_{N(J)}$ is the complementary projector, i.e., an operator which projects the vector δq_0 to the nullspace of Moore-Penrose generalized inversion $J^+(q)$ of the Jacobian matrix.

Homogeneous member in (5) is the formal vehicle which can be used for generation of internal motions in the robot mechanism, i.e., the selfmotions which do not alter the robot TCP position. Since the jointspace stiffness matrix is dependant of the robot joint coordinates, (3) and (4), internal motions in the Jacobian nullspace will affect the jointspace stiffness matrix. In that sense, nullspace motion can be used to satisfy secondary task objective, [8, 7], that is control of generalized stiffness of the robot TCP.

In accordance to [8] and [9], the complementary projector used in (5) is defined by the following relations:

$$P^c_{N(J)}(q) = I_n - P_{N(J)}(q), \quad (6)$$

$$P_{N(J)}(q) = J^+(q) J(q) \quad (7)$$

while the Moore-Penrose generalized inverse satisfying the least norm condition is given by the following relation, [18]:

$$J^+(q) = J^T(q)[J(q)J^T(q)]^{-1} \in R^{n \times m}. \quad (8)$$

Relations (5) to (8) are essential for the proposed Configuration-based Stiffness Control method (CFC).

3.2 The Cost Function

According to the relation (5), an arbitrary joint displacement vector δq_0 should be determined in a way to satisfy a suitable cost function $u(q)$. The cost function $u(q)$ should be defined as an appropriate measure of the discrepancy between the desired robot TCP stiffness K_{xd} and induced robot TCP stiffness K_x. In our approach we have focused on the robot jointspace and synthesis of the cost function which is related to the robot jointspace stiffness matrix, following the basic relations for its analytical definition, given in (3) and (4). This leads to the following definition of the nullspace vector optimization criterion:

$$\delta q_0 = \{\delta q_0 \in R^n : \delta q_0 = \min(\Delta K_q), \Delta K_q = K_{qd} - K_q\} \quad (9)$$

This leads to the following proposition: Euclidian norm of the off-diagonal elements of the jointspace stiffness matrix, which is analytically defined by the following relation:

$$u(q,K_x)=\left\|k_{q\_ij}(q,K_x)\right\|=\sqrt{\sum_{i=1}^{m-1}\sum_{j=i+1}^{n}(k_{q\_ij}(K_{xd},q))^2} \quad (10)$$

is a sufficiently representative measure of the discrepancy between the desired robot TCP stiffness and the robot TCP stiffness induced by the monoarticulated linkage of the robot mechanism, and can be effectively used as a cost function for the robot TCP point stiffness optimization, consistent with the Configuration-based Stiffness Control framework (CSC), formally based on relation (5), and the set of constraints c1 to c3.

The cost function (10) generates a scalar potential field over the robot hyperdimensional configuration space. Since this potential field is nonlinear, continuous, and therefore differentiable function, the gradient optimization method, [8, 7] can be effectively applied to find the optimal nullspace joint vector q_0* which locally minimizes influence of the cross-joint members of the robot jointspace stiffness matrix. This leads to the following relation:

$$\delta q_0=-\alpha\nabla u(q)=-\alpha\frac{\partial}{\partial q}u(q),\quad \alpha>0\,. \qquad (11)$$

The scalar operator, $\alpha > 0$, is used to scale the magnitude of the δq_0 and is defined as follows:

$$\alpha(t_k)=\beta\frac{\left\|\delta q_P(t_k)\right\|}{\left\|\delta q_N(t_k)\right\|},\beta>0\,, \qquad (12)$$

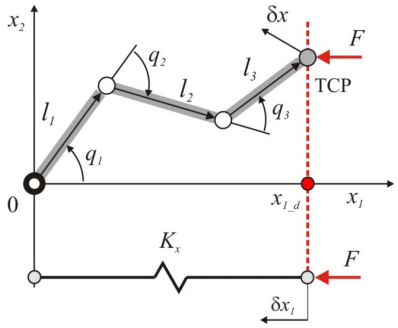
where β is the global multiplier, which is time independent and defined by the supervisor, or by the higher levels of the robot control system.

4 SIMULATION EXAMPLE

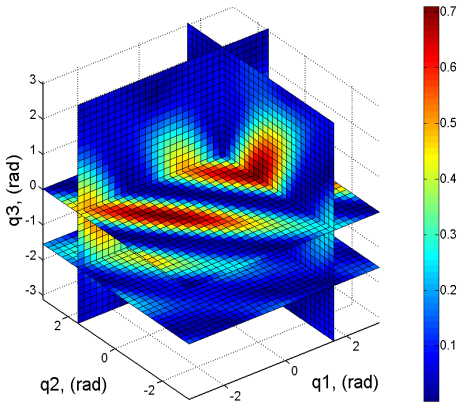
The effectiveness of the proposed CSC framework was verified by computer simulations using planar, Minimal Redundancy Robot arm with two redundant d.o.f. – MRR-R2 robot. Model of the robot arm and its taskspace is shown in Fig. 2a. Volume potential field which is generated by the cost function (10) and the adopted set of link parameters: l_1=0.360m, l_2=0.360 and l_3=0.275m is shown in Fig. 2b. In order to depict its internal dynamics along the q_1 direction, an orthogonal slice at the arbitrarily chosen point $q_1=\pi/2$ is given in Fig. 9c. The field segment shows that the adopted potential field is generally well-behaved. The corresponding streamlines are smooth and strongly converge to local minima.

Capacity of the MRR-R2 linkage to generate the desired TCP stiffness is illustrated graphically in Fig. 3c, which shows the same potential field of MRR-R2 as in Fig. 2b, but reduced to the subspace $u(q, K_x) \le u_B$ where u_B is a boundary isopotential surface satisfying $u_B = 0.03(\max(u(q, K_x))$. This subspace corresponds to the configurations which induce TCP stiffness very close or equal to the desired value, and is conditionally related to the subspace Q defined in the section 3.

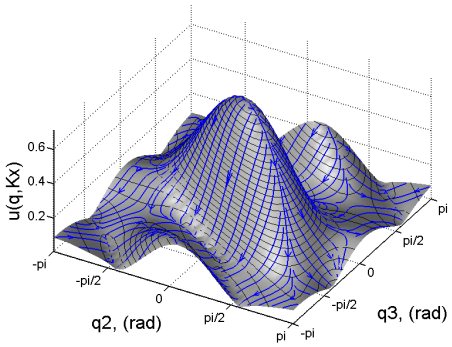
a)



b)



c)



d)

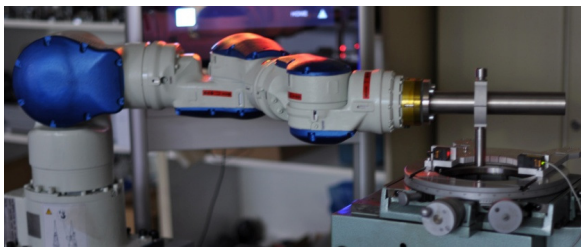


Figure 2. *Model of the minimal redundant robot MRR-R2 in the taskspace (a) and the corresponding volume potential field (b). Potential field dynamic along the q1 direction for slice plane q1=π/2, with the corresponding streamlines (c). Yaskawa SIA-10F, 7 d.o.f. robot arm used in physical simulations (kinematically reduced to 3 d.o.f. in horizontal plane to avoid influence of gravitational terms), (d).*

A series of simulation experiments were performed in order to prove the functionality of the proposed CSC framework

with MRR-R2 robot arm. As an example, Fig. 3a and 3b show the results of the simulation of the case when both tasks were active and when the value of the β multiplier was set to 2 in order to speed up the convergence of the secondary task toward the optimal value of the TCP stiffness. In both tasks the operators converge steadily to optimal/desired values. However, it is obvious that after the convergence of the secondary task (iteration k=305) dithering motions occur, which significantly influence the robot TCP, causing crosstalk between the secondary and the primary task. Such behaviour is observed when the secondary task reaches its optimal value, when the nullspace motion generator becomes unstable due to sudden changes in the gradient sign (11). This problem could be partially solved by low-pass filtering. However, a more comprehensive solution requires introduction of additional secondary tasks, first of all the task which avoids singularities, [9 and 10].

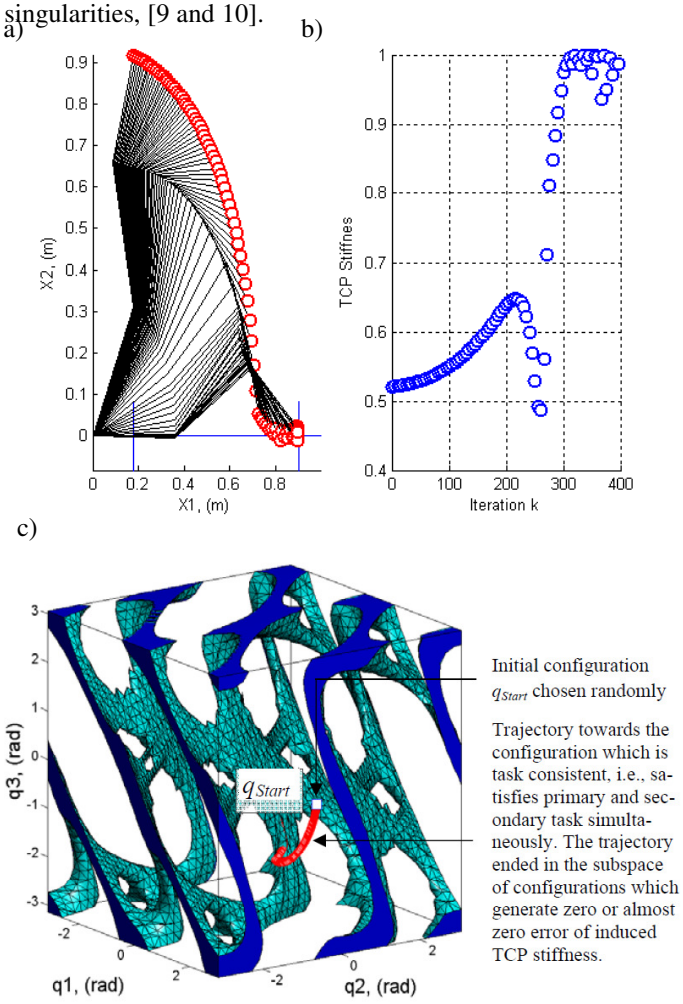


Figure 3. *Simulation results for the MRR-R2 robot motion from x0= 0.180m up to xd= 0.900m, and execution of the secondary task – optimization of the TCP stiffness: a) Robot mechanism motion in the operational space; b) TCP stiffness variation; c) subspace of the scalar potential field satisfying u(q,Kx)< 0.03[max(u(q,Kx))], extracted from Fig. 8b, and robot motion in the configuration space.*

6 CONCLUSION

In this paper, the Configuration-based Stiffness Control (CSC) method is formulated as a kinetostatically consistent framework for robot compliant behaviour control, based on gradient projection of the cost faction which minimizes the norm of off-diagonal elements of the jointspace matrix. We have found that in robotic arms which are driven by variable stiffness monoarticular actuators, kinematic redundancy and nullspace of the corresponding Jacobian matrix are potentially applicable for effective control of robot TCP compliant behaviour by canonization of the jointspace stiffness matrix. However, singularity avoidance and joint range limits must be taken into account as an additional secondary task.

References

[1] Hogan, N.: Impedance Control: An Approach to Manipulation, Part I, II and III, ASME Transactions, Journal of Dynamic Systems, Measurement, and Control, Vol. 107, pp. 1-24, 1985.

[2] Khatib, O.: An Unified Approach to Motion and Force Control of Robot Manipulators: The Operational Space Formulation, IEEE Journal of Robotics and Automation, Vol. 3, No. 1, pp: 42-53, 1987.

[3] Salisbury, J.: Active Stiffness Control of a Manipulator in Cartesian Coordinates, 19th IEEE Int. Conference on Decision and Control, pp. 95–100, 1980.

[4] Albu-Schaffer, A., et al.: The DLR lightweight robot: Design and control concepts for robots in human environments, Industrial Robot: An International Journal Vol. 34, No. 5, pp. 376–385, 2007.

[5] Khatib, O.: Motion/Force Redundancy of Manipulators, *Symposium on Flexible Automation*, Japan-U.S.A., pp: 337-342, 1990.

[6] Chen, S.F., Kao, I.: Geometric Approach to the Conservative Congruence Transformation (CCT) for Robotic Stiffness Control, *IEEE International Conference on Robotics and Automation*, pp. 544–549, 2002.

[7] Liegeois, A.: Automatic supervisory control of the configuration and behavior of multibody mechanisms, IEEE Transactions on Systems, Man and Cybernetics, Vol. 7, No. 12, pp: 868-871, 1977.

[8] Baillieul, J., Hollerbach, J., and Brockett, R.W.: Programming and Control of Kinematically Redundant Manipulators, *23-rd IEEE Conference on Decision and Control*, Las Vegas, USA, pp. 768-774, 1984.

[9] Yanai, H., Takeuchi, K., and Takane, Y.: *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*, Springer Science+Business Media, ISBN 978-1-4419-9886-6, 2011.

[10] Baerlocher, P, Boulic, R.: An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels, The Visual Computer, Vol. 20, No. 6, pp. 402-417, 2004.

ON DEVELOPING ANTHROPOMIMETIC ROBOT-ARM

Aleksandar Rodić, Branko Miloradović, Đorđe Urukalo
Robotics Laboratory
Mihajlo Pupin Institute
Volgina 15, 11060 Belgrade, Serbia
Tel: +381 11 674236; fax: +381 11 6776 583
e-mail: aleksandar.rodic@pupin.rs

ABSTRACT

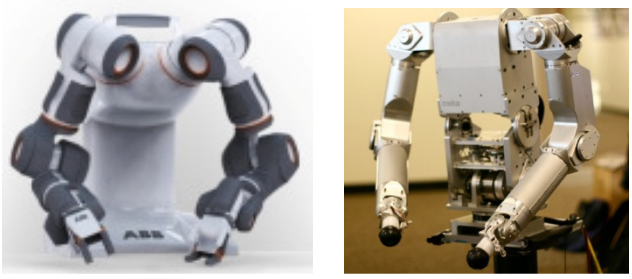
Mechanical structure of an anthropomimetic lightweight robot arm is considered in the paper with aim to achieve technical performances and appearance closer to the physical properties and capabilities of biological limb. New mechanical structure of the arm is proposed in the paper that takes into account to a greater extent biological aspects (physiological and structural). A redundant 7 DOFs, tendon-driven, biologically-inspired robotic arm with poly-articulated joints characteristic for an over actuated system is presented in the paper. Compliant structure that needs to enable safe and reliable human-robot interaction was achieved by introducing additional passive elasto-damping elements into the system and by means of control software that change system impedance in a biological way. Corresponding control-block scheme and CAD-model of the robot-arm mechanical structure will be presented in the paper, too.

1 INTRODUCTION

A human arm with hand has an exceptional dexterity and physical capabilities (weight carrying, doing impact, pulsation, etc.) by virtue of its own naturally adjusted structure of the musculoskeletal system, sensitive perceptive system and fast and reactive nervous system whose peripheral terminations reach up the every segment of the limb. By virtue of such a sophisticated structure, the arm enables humans accomplishing the most complex and delicate manipulative tasks that require high speed and precision, perceptive capabilities, but also natural robustness and durability of operation even under load. Contemporary service robots use today the *lightweight robotic-arms* (LWA) due to intention to make robot arm/hand as similar as possible to the human limb by its structure and functionality, in sense of decreasing the weight of the arm, increasing the payload up to the human physical capabilities, achieving desired compliance of joints (soft joints), decreasing energy consumptions for operation, etc. Industrial lightweight robot arms rely to the contemporary technology, miniature powerful actuators, miniature embedded electronics, new light and durable composite materials, harmonic-drives, integrated mechanics and electronics, etc. In spite of that, the overall mass of the lightweight robots is still higher than a human limb and also the payload fraction (coefficient that quantifies robot weight and payload ratio) overcome those that is with human arm. Due to these facts the energy consumptions are still high because of the actuator imperfectness in spite of the high progress of technology. Bearing in mind the fore mentioned, this paper represents an attempt to make a next step towards a new concept of design lightweight robot arms of anthropomorphic characteristics.

2 STATE-OF-THE-ART

The lightweight robots is mainly intended for industrial applications which demand high accuracy and repeatability (up to ± 0.1mm), payload (up to 10kg) and composite speed of 3-5 m/s. These systems also have high stiffness and weights from 15 up to 35 kg and it also needs protected work area in order to prevent collision with objects and people in its environment. These manipulators have stiff joints without compliant. World famous lightweight robotic manipulation systems are: KUKA KR Agilus 6 R700 [1], Shunk LWR Powerball [2], Mitsubishi LWAPA10 [3], Yaskawa-Motoman SIA5F [4], Denso VS-6577G-B [5], Barett robotic arm [6] etc. Exception in this case gives new outstanding KUKA LBR iiwa 7R800 [7,8] lightweight robotic arm, which satisfies the most rigorous technical criteria and also has soft joints which enables human robot interaction in the workspace of the robot. This robotic system represents an intermediate form of above mentioned types of LWR (industrial and service/personal).
Parallel with permanent progress in development of LWAs dedicated to industrial purposes, intensive work on design bi-manual, i.e. dual-arm manipulation systems for humanoids and service robots was accomplished in the recent years. Two technologically very advance realizations can be mentioned here: (i) the FRIDA dual arm system [9], and the Hizook Meka robot [10] developed at the MIT presented in Fig. 1.



a) b)

Figure 1: *a) Dual-arm concept-robot FRIDA [9]; b) The Meka A2 compliant manipulator with 7-DOFs [10]*

3 BIOLOGICAL ASPECTS OF LWA DESIGN

By analyzing existing LWR manipulation systems presented in Section 2, it can be concluded that the primary goal with such systems is to achieve relatively lightweight structure, fine technical performances such as high accuracy and repeatability of the arm, high composite (end-tip) speed and structural compliance in terms of existence "soft joints" which ultimately provides safe interaction with humans (coworkers). The advanced LWA features are achieved by miniaturization of the structural components (sensors, actuators, etc.). Compliant structures are achieved by introducing harmonic-drive or serial elastic actuators combined with force (torque) feedback in robot joints. General impression is that despite the undeniable progress in the design of the LWAs, the anthropomorphism degree as an important indicator, is sacrificed and is still far away from the biological system. The term anthropomorphism used in the paper describes the property of a system that is a measure of the degree how the technical system applies biological methods and principles for its' operation as well as how it looks like the biological system as referent model.

The term, „degree of anthropomorphism" or simply „anthropomorphism" is introduced when plan to design a bio-inspired system such as for example anthropomorphic robot arm. Anthropomorphism represents the measure of how much the machine, in this case it is an artificial i.e. robot arm, is similar to the human limb and to which extent the principle of operation (functioning) of technical system mimics the methodology of the natural behavior of the reference biological model. Anthropomorphic robot-arm ensures the following advantages with respect to the conventional industrial LWR: (i) appearance and functionality better fit model of the biological limb, (ii) better social acceptability in tasks of direct interaction with human users/partners, (iii) energy saving, (iv) better ergonomic performances, etc. The main attributes of an anthropomorphic robot-arm are as to follow: (i) redundant kinematical structure with 7 degrees of freedom (DOFs), (ii) existence of the functional spherical joint in the shoulder instead of three cylindrical sequentially ordered joints, (iii) polyarticular joint activation characteristic for the over actuated systems instead conventional monoarticularly actuated mechanical structure, (iv) linear actuators use as the replacement of natural muscles according to the model of human arm, (v) synergy of complementing actuators of a polyarticular joint aiming to energy saving, even payload distribution amongst the actuators, and taking over the risk in a case of failure of one within the complementing pair, (vi) light mechanical structure converging to the human arm weight, (vii) payload fraction (weight-payload ratio) close to value 1:1 that is characteristic to humans.

The degree of anthropomorphism in the case of robot arm cannot be measured but can be estimated based on the indicators explained in the previous paragraph. The objective of the research presented in this paper regards to design of a conceptually novel mechanical structure of the LWA based on the set of anthropomorphic criteria imposed even at the cost of some degradation of technical performances in order to achieve system of more anthropomorphic properties.

The evolution took thousands of years during which the human body underwent significant changes in accordance with the conditions and way of life. Thus, the human hand and arm, as the body part used for work, have suffered significant morphological and functional changes. A human arm has a complex structure consisting of upper arm and two forearm bones and large number of muscles (Fig. 2).

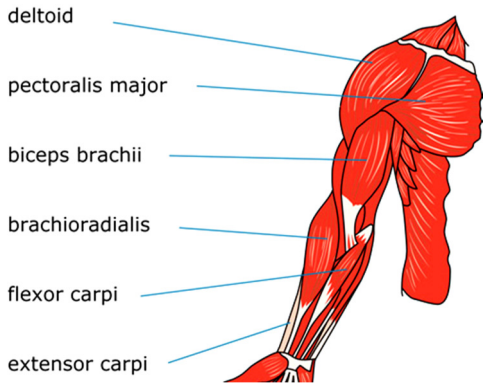


Figure 2: *Anatomy of human arm – muscular system*

Human arm has a larger number of muscles than the number of joints. Movement of the arm joints is realized by the synergy of corresponding muscles. In this way, the nature enabled arm to move even if some of the muscles are injured or if its functionality is disabled for some other reasons. At the same time, the payload on the arm is distributed amongst muscles. That increases stability and robustness of the system so that physical fatigue upon a muscle is compensated by additional effort (activity) of its complement. Also, in some "non-demanded" tasks, it is not necessary all muscles to be simultaneously activated. Some of them are relaxed while the rest ones are active. This natural principle of payload distribution in the particular joints is implemented in the robot - arm design in this paper. The basic arm movements are: adduction/abduction, flexion and extension, pronation and supination as well as interior and exterior rotation (Fig. 3).

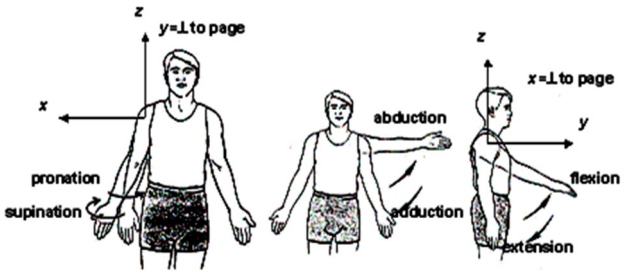


Figure 3: *Basic arm movements articulated by different groups of arm muscles.*

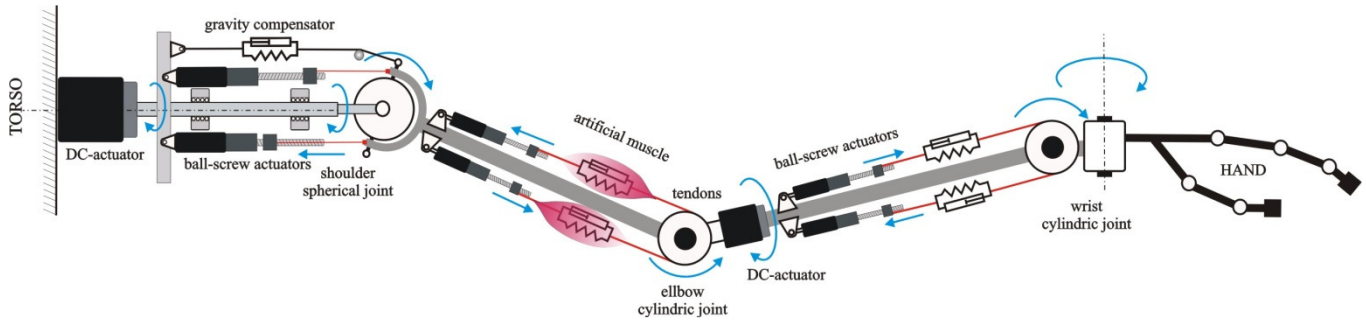


Figure 4: *High-level system description - principle of actuation, compliant elements, spherical joint in the shoulder.*

4 CONCEPT AND DESIGN OF ROBOT-ARM

Considering that the aim of the research presented in this paper is development of a robotic arm with a high degree of similarity and functionality with a biological model, we set the following criteria to be met:

- Repeatability of the robot-arm (1 mm),
- End-tip speed of the robot-arm (3 m/s), and
- Payload in position of full abduction, Fig. 3 (3 kg).

In kinematic sense, the proposed mechanism of robotic arm represents a redundant structure with 7 DOFs which allows fine mobility inside the work space that corresponds to the human-arm range. For such a mechanism, it is planned 6 actuators for upper arm rotation in shoulder joint, 4 actuators for the forearm in elbow joint and 4 actuators for the corresponding rotations in the wrist joint. This amounts to a total of 14 actuators which drive 7 degrees of freedom. Pronation and supination of the upper arm (Fig. 3) are enabled by one DC motor that rotates the entire spherical wrist with a set of five quality spindle drive actuators which minimize the effects of friction. The shoulder joint has the appropriate amenability (compliance) set through passive elastic-damping elements (springs and dampers) that have a role to compensate for gravitational torques in the wrist, to allow proper compliance of the wrist when overloaded. In a way these elements are an imitation of muscles in the system. At the same time, these elements act as antagonists of the elements that tend to make a recurrent movement of the wrist in flexion / tension and abduction / adduction. Abduction and adduction movements are achieved by activating the corresponding pairs of spindle-drive actuators, where the two motors are employed for abduction and one (with the interplay of artificial muscle, Fig. 4) for adduction arm. It should bear in mind that gravitational torques furthering hand adduction in this case. Regarding to flexion and extension arm movement, there are similarities with the previous case. In this paper we accepted the same two linear spindle-type motor drives, one for bending and one for elongation.

In the elbow joint, as possible movements, there is a forearm pronation/supination as well as flexion and extension movements. Pronation and supination are provided with a rotary DC motor, while bending and straightening is enabled with the assistance of a pair of linear actuators: 2 motors for flexion and 1 motor for extension movement. In the robotic forearm, micro actuators with ball screw were placed radially. The number of DOFs corresponds to the DOF of the hand plus 2 engines for flexion/extension and abduction / adduction hand movement in wrist joint. Motors types and their performances are determined based on hand model simulation for conditions that correspond to the set criteria mentioned in the preceding section. The principle of actuation is simplistically depicted in a 2D diagram in Fig. 4. The CAD model of the proposed mechanical design is presented in Figs. 5 and 6.

5 CONTROL OF ROBOT-ARM

Controller of the robot arm consists of an impedance controller and an intelligent joint torque controller (Fig. 7). Within the impedance controller, there are two feedbacks: position/speed feedback of X and force/torque feedback of F in the center of mass of the hand. The impedance controller determines the control torques T_0 in certain hand joints that provide the desired movement and the system impedance. Since the system is over-actuated, intelligent torque controller has to make appropriate distribution of control loads on some engines that strengthen the whole system. Being the 7 DOFs system has 14 motors, intelligent control algorithm determines load distribution in order to save energy and minimizes the load on several engines. The algorithm takes into account the experimental experience on human muscles and load distribution in biological systems. Output variables from torque controllers in joints are reference torques τ_I in individual servo-actuators of the system. The joint torque controller uses torque feedbacks T in certain hand joints.

The robot arm and hand dynamics are coupled and presented in the scheme with appropriate feedbacks shown by dashed lines in Fig. 7. Torques of the hand are reduced to its center of mass and presented by a vector T_h. The proposed control scheme allows compliance of the system by the effects of the external forces and torques on the system, or in other words, compensates torques generated as a disturbance in the system caused by human action or collision with objects in the environment.

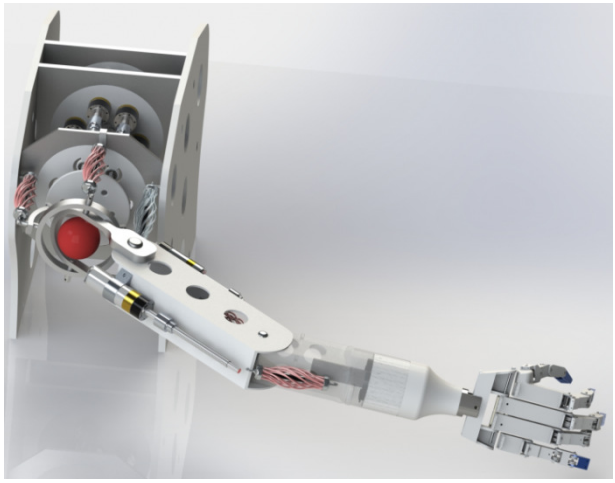
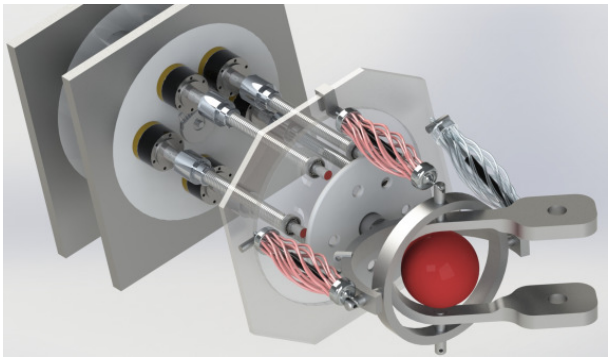
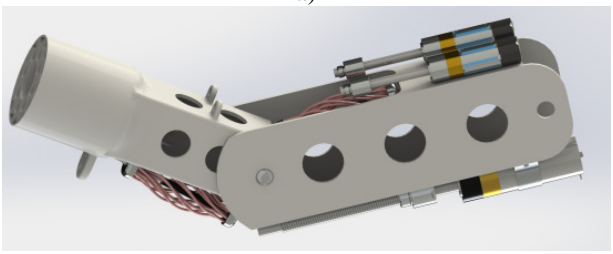


Figure 5: *CAD model of the 7DOFs lightweight anthropomorphic arm with 20 DOFs robot-hand*



a)



b)

Figure 6: *Details of the robot arm: a) actuation of the spherical shoulder joint; b) upper arm and forearm.*

The intelligent torque controller takes into account the disturbance torques T_d and on its output generate new control signal τ_I that is the result of compromises between the desired movement and disturbance generated due to collisions with the environment.

6 CONCLUSION

The following conclusions can be derived: (i) There is no better design of the arm to be made than the nature has done through the evolution; (ii) Robot arm to be designed should not be necessary more accurate/precise than human model (up to 1.5 mm); (iii) Robot hand ensures high accuracy of the system not arm (0.1 mm); (iv) Over actuated robot arm with polyarticulated joints (shoulder and elbow) has significant advantage over the conventional monoarticulated systems; (v) Compliant structure is inevitable design concept in order to achieve anthropomorphic functionalities; (vi) Mechanical structure with spherical joint in the shoulder decreases the overall mass of the system but increase the kinematical complexity; (vii) Over actuated robot arm structure saves energy due to the inherent capability of planning payload distribution per actuators.

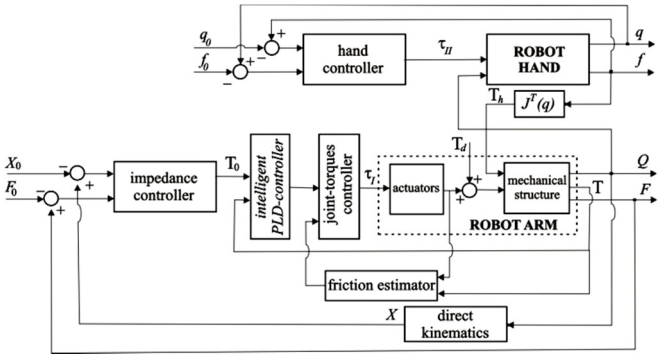


Figure 7*: Control-scheme of anthropomorphic robot-arm.*

References

[1] http://www.kuka-robotics.com/res/sps/e6c77545-903049b193f54d17c92173aa_Spez_KR_AGILUS_sixx_en.pdf

[2] http://www.schunk.com/schunk_files/attachments/LWA_datasheet_2012-03__EN.pdf

[3] https://www.mhi.co.jp/technology/review/pdf/e405/e405298.pdf

[4] http://www.motoman.com/datasheets/SIA5F.pdf

[5] http://www.densorobotics-europe.com/en/product/vs-6577g-b

[6] http://www.barrett.com/robot/products-arm.htm

[7] Rainer Bischoff, Johannes Kurth1, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald2, Gerhard Hirzinger, The KUKA-DLR Lightweight Robot Arm – a new reference platform for robotics research and manufacturing, Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)

[8] KUKA LBR iiwa 7R800 www.kuka-labs.com/en/pressevents/news/201304_LBR_iiwa_world_premiere.htm

[9] Meka Robotics' Humanoid Torso and Anthropomorphic Hands, Meka Robotics A2 Compliant Manipulator.

[10] http://www.abb.com/cawp/abbzh254/8657f5e05ede6ac5c1257861002c8ed2.aspx

TAKS-SPECIFIC DYNAMICS FOR COMPLIANT ROBOT BEHAVIOR AND ACCURATE TRAJECTORY TRACKING

Miha Deniša, Andrej Gams and Tadej Petrič
Department of Automation, Biocybernetics and Robotics
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773140; fax: +386 1 4773154
e-mail: miha.denisa@ijs.si

ABSTRACT

The main advantage of using the feed-forward task-specific dynamics is to ensure properties like accurate trajectory tracking and compliant robot behavior. Without using task-specific dynamics, accurate trajectory tracking could only be achieved using high-gain feedback control, which is not compliant. Vice versa, low-gain feedback control will behave compliantly, but cannot accurately track the desired trajectory. To combine both features into a single control approach, we propose to use a multi-step process where in the first step a high-gain feedback control is used to learn the task-specific dynamic model, and then in the second step the task-specific model is used in a feed-forward manner, but with a low gain feedback. Combining low-gain feedback with feed-forward task-specific dynamic model enables compliant robot behavior and accurate trajectory tracking. In this paper we compare the performance of the proposed approach using feed-forward task-specific dynamics, with the classical feedback control approaches.

1 INTRODUCTION

Different algorithms for machine learning were proposed and later adopted in robotics, mainly because of their ability to learn complex mathematical models. It was shown that they can successfully learn the robot kinematics [1], dynamics [2], [3], or even robot locomotion [4]. Even though these learning algorithms can learn such complex models, the necessity to acquire large training data remains their main drawback. To improve learning and system performance, biologically inspired methods, were proposed for robot control. An extensive review was recently published by Franklin and Wolpert [5].

In this paper we propose a control framework where task-specific dynamics is learned in the first step using high-gain feedback control. In this step the robot is stiff and it is accurately tracking the desired trajectory. Because of the high stiffness it can only operate in a well-defined environment under human supervision. After the learning phase, the robot can perform the same task using feed-forward task specific dynamics, but with low feedback gains, which ensures compliant behavior. In case of an unexpected collision, the compliant behavior will ensure low impact forces. This allows that the robot can now work in unstructured environment or with humans, because the danger of damaging itself, the environment, or people around it is significantly reduced through compliant behavior.

To implement the proposed control structure, we use Dynamic Movement Primitives (DMPs) – a method of encoding a trajectory with a set of differential equations and a weighted combination of kernel functions, described by Ijspeert et al. [7]. By encoding both the spatial and torque trajectories, the proposed system structure makes advantage of the stability properties of the DMPs as well as of their modulation and generalization properties.

The paper is organized as follows. In section 2, we give a detailed description of the proposed learning system. In section 3 we evaluate the proposed approach. Conclusions and summary are given in section 4.

2 CONTROL FRAMEWORK

For a robot with rigid bodies, the equations of motion are given as

$$\mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\epsilon}(\mathbf{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) = \boldsymbol{\tau}, \qquad (1)$$

where $\boldsymbol{q},\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ are the joint position, velocity and acceleration, $\mathbf{H}(\boldsymbol{q})$ is the inertia matrix, $\mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ are the Coriolis and centripetal forces, $\boldsymbol{g}(\boldsymbol{q})$ is the gravity vector and $\boldsymbol{\epsilon}(\mathbf{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ are the nonlinearities not considered in rigid body dynamics. The inverse dynamic model of the robot is denoted as $\boldsymbol{f_d}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$. To track the desired trajectory $\boldsymbol{q_d}$, a possible solution is

$$\boldsymbol{\tau_u} = \mathbf{K}(\boldsymbol{q_d} - \boldsymbol{q}) + \boldsymbol{D}(\dot{\boldsymbol{q}}) + \boldsymbol{f_d}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}), \qquad (2)$$

where **K** and **D** are the diagonal matrices that define the stiffness and the damping, respectively. If the elements of matrix **K** are high, the robot is stiff and it is accurately tracking the desired trajectory. Vice versa, if the elements of the matrix **K** are low, the robot is compliant, but it cannot

accurately track the desired trajectory. To have both properties, i.e. accurate tracking and compliant behavior, we proposed a controller with task-specific feed-forward term,

$$\boldsymbol{\tau_u} = \mathbf{K}(\boldsymbol{q_d} - \boldsymbol{q}) + \boldsymbol{D}(\dot{\boldsymbol{q}}) + \boldsymbol{f_d}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) + \boldsymbol{\tau_f}, \quad (3)$$

where $\boldsymbol{\tau_f}$ is the task-specifc feedforward signal, that compensates the task-specific dynamics. Different options exist for obtaining task specific dynamics. One possibility is to mathematically model the task, as it is usually done for robot dynamics. Note that robot dynamics is also included in the control. However, to model task-dynamic in such a way, it would become a very time consuming modelling operation that would have to be performed even for the smallest of interaction tasks by expert. Moreover, it cannot be expected form a regular user to perform such modeling. Therefore we propose a solution where the robot learns the task-specific dynamics by itself in a controlled environment under human supervision. Once a robot learns the task-specific dynamics it can execute the desired task accurately while at the same time being compliant. In case of an unexpected collision, compliant behavior implies low impact forces. This further means that the robot can now safely work in unstructured environment, or with humans.

After the desired motion trajectory $\boldsymbol{q_d(t)}$ is executed using a classic feed-back controller, corresponding torque signals $\boldsymbol{\tau_f(t)}$ are gained. The task is now defined by this pair of signals

$$\boldsymbol{h(t)} = \left[\boldsymbol{q_d(t)}, \boldsymbol{\tau_f(t)}\right]. \quad (4)$$

To encode both of them, the principle of Dynamic Movement Primitives (DMPs) is used. While motion trajectories are encoded as DMPs, corresponding torque signals are encoded with Gaussian kernel functions.

A short overview of the DMPs is given next. The equations are valid for one degree of freedom (DOF). For multiple DOFs the equations can be used in parallel. The nonlinear system of differential equation defines a DMP. The system is given as

$$\nu\dot{z} = \alpha_z(\beta_z(g - y) - z) + f(s), \quad (5)$$

$$\nu\dot{y} = z, \quad (6)$$

Where the linear part ensures the convergence of y to the desired goal configuration g and the nonlinear part $f(s)$ modifies the shape. It is defined by a linear combination of M radial basis functions given [6] by

$$f(s) = \frac{\sum_{i=1}^{M} w_i \psi_i(s)}{\sum_{i=1}^{M} \psi_i(s)} s, \quad (7)$$

where ψ_i denotes Gaussian basis function

$$\psi_i(s) = \exp(-h_i(s - c_i)^2). \quad (8)$$

Here, c_i denotes the centers and h_i denotes the width of the Gaussian basis functions. If proper parameters, i.e., $\alpha_z = 8$, $\beta_z = 2$ and $\nu > 0$, are chosen the system will converge to desired configuration.

To encode the desired trajectory the target function given as

$$f_t = \nu^2\, \ddot{q}_d(t) + \alpha_z \dot{q}_d(t) - \\ -\alpha_z\beta_z\big(q - q_d(t)\big), \quad (9)$$

is learned by applying the locally weighted regression.

The corresponding task-specific torques are obtained by executing the trajectory encoded as DMP using a high gain feedback controller, which ensures the required tracking accuracy. The corresponding torques τ_f are then encoded as a linear combination of basis functions

$$\tau_f = f(s). \quad (10)$$

The target function, calculated from Eq. (9) for task specific toques is now

$$f_t = \tau_x(t), \quad (11)$$

where $\tau_x(t)$ is the torque of the high gain feedback control. Again, locally weighted regression is used to learn the target function and encode corresponding torques $\boldsymbol{\tau_f(t)}$.

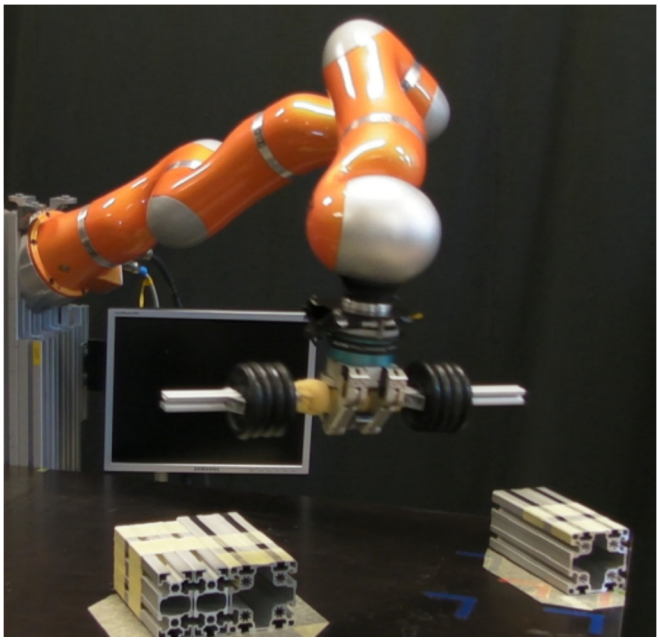


Figure 1: Experimental setup where the robot picks and places an object.

3 EVALUATION

To evaluate the proposed control approach we use a Kuka LWR robot with a BarrettHand. The goal was to pick up and place down an object with different weights. The experimental setup is shown in Figure 1.

First we analyzed the effect of stiffness parameter on the tracking error with and without using stiffness feed-forward task-specific torque signal. Human coach initially demonstrates the motion to the robot, that moves an object form its initial position to the final position. The movement was then executed three times using high gain feedback control, which ensures required tracking accuracy. For each execution we change the mass of an object in steps of 2 kg, and corresponding torques were learned.

Next the motion encoded as DMPs was executed with varying stiffness, using only feedback control. Then, the motion was executed again, but this time with the learned feed-forward torques. To compare the performance of each controller the maximum error for each task execution was defined as

$$e_m = \max(||\boldsymbol{p}(t) - \boldsymbol{p_d}(t)||). \qquad (12)$$

Where $\boldsymbol{p}$ is the measured task space position of the end-effector, and $\boldsymbol{p_d}$ is the desired task space position in vector space. The results are shown in figure 2, where orange denotes the control without using feed-forward task dynamic and the blue denotes the control with learned feed-forward task dynamics. We can see that the tracking error is statically significant if feed-forward task specific torque is not used compared to the proposed approach. We can also see the point where the errors start to notably increase, i.e. until really low values of stiffness.

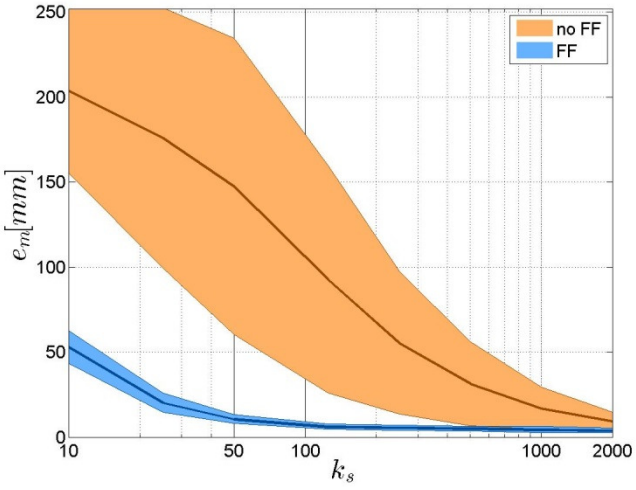


Figure 2: Mean and standard deviation of tasks maximum error. The blue color line shows the maximal mean and standard deviation for the proposed system with task-specific feed-forward torques. The orange line shows the mean and the area the standard deviation for the feedback control without feed-forward toque signal.

Based on these results we can conclude that the tracking error is marginal if the stiffness is above 50, where 50 denotes a very low stiffness, as defined in the robot's controller. To evaluate the impact with an object while using the proposed approach we conducted an experiment, where the robot unexpectedly collides with an environment. Here our proposed approach was compared with feedback once with high gain and next with low gain. The snapshots of the results are shown in Figure 3, were we can clearly see in the first pictures, that if the robot is using high gain control it will try to compensate for the error and therefore it will penetrate into an object. On the other hand, with low feedback, the impact force will be small but the robot will not be able to track the desired trajectory as can also be seen in Figure 4. In this particular example, if tracking would be accurate, the contact point with the object should be on the edge. Finally, for the proposed control, we can see that the tracking is accurate and the impact force is lower.

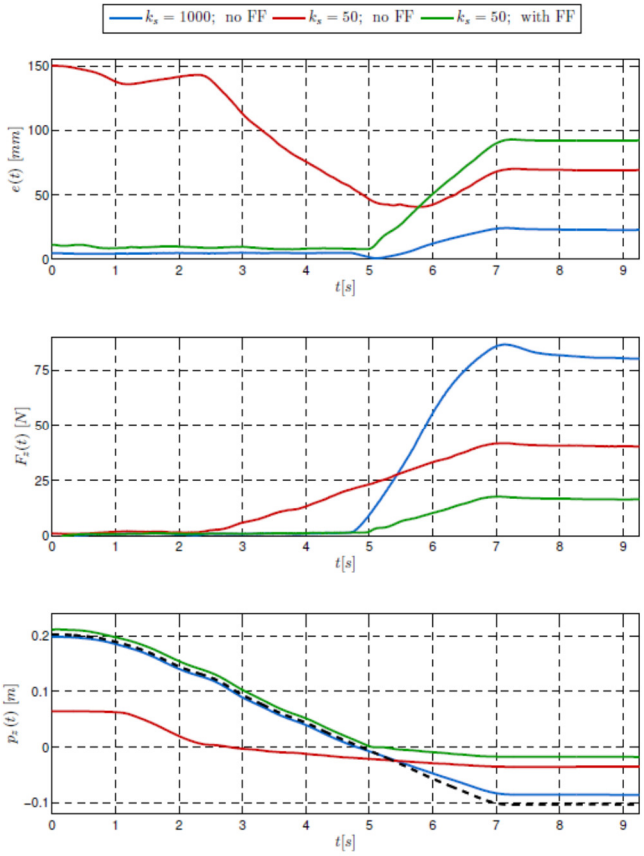


Figure 4: Robot colliding with an object with different stiffness settings and control approaches. The graphs present collisions under two different stiffness settings (1000 and 50) and two different control approaches. One uses the feedback control, while the other adds learned feed-forwarded torques. The top graph shows position errors, the second one forces at the end effector in z axis and the bottom one actual robot positions in the significant dimension with the object starting at 0 m.

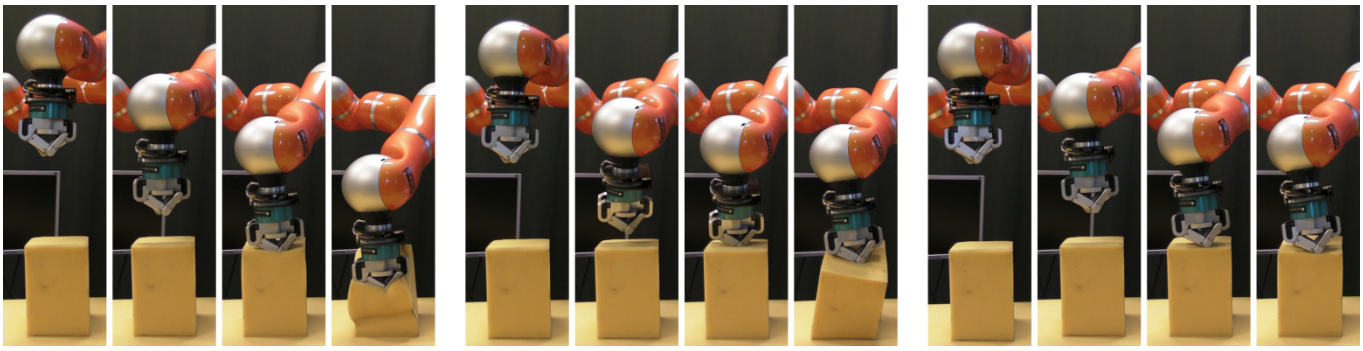


Figure 3: Robot colliding with an object with different stiffness settings and control approaches. In the first series of images the robot moved in the stiff manner while the middle images show a collision while executing compliant movement. Both of the movements were executed while using a standard feedback loop control. The last series of images show the collision while moving in a compliant manner and using the proposed control with feed-forwarded torques.

6 CONCLUSION

We proposed a new task-specific dynamic learning framework which can autonomously learn proper motion trajectories and the corresponding task dynamics to perform a desired task. The learning of motion patterns with DMPs and of task-specific dynamics with Gaussian kernel functions is done in real time and without any additional signal processing methods. As such, the proposed learning framework enables simple and computationally inexpensive control in the case of dynamically challenging tasks. The main contribution of our approach the learning the task specific dynamics which ensures accurate task execution and at the same time compliant robot behavior. In the future, we would like to generalize the approach for the periodic tasks and incorporate statistical learning methods to build a library of motions including task-specific dynamics.

References

[1] a. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2001, vol. 1, no. Iros, pp. 298–303.

[2] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model Learning with Local Gaussian Process Regression," *Adv. Robot.*, vol. 23, no. 15, pp. 2015–2034, Jan. 2009.

[3] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey.," *Cogn. Process.*, vol. 12, no. 4, pp. 319–40, Nov. 2011.

[4] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 167–172.

[5] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control.," *Neuron*, vol. 72, no. 3, pp. 425–42, Nov. 2011.

[6] A. Gams, T. Petric, A. Ude, and L. Žlajpah, "Performing Periodic Tasks: On-Line Learning, Adaptation and Synchronization with External Signals," in *The Future of Humanoid Robots - Research and Applications*, R. Zaier, Ed. Intech, 2012, pp. 1–28.

[7] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," Neural Computation, vol. 25, no. 2, pp. 328–373, 2013.

Indeks avtorjev / Author index

Batinica 13
Bevec 5
Borovac 13, 17
Danilov 25
Deniša 33
Gams 9, 33
Lukić 25
Miloradović 29
Nemec 21
Nikolić 13, 17
Petrič 21
Petrović 25
Raković 13, 17
Rodić 29
Savić 13, 17
Ude 21
Urukalo 29

Konferenca / Conference
Uredili / Edited by

Robotika /
Robotics
Jadran Lenarčič, Aleš Ude, Bojan Nemec, Andrej Gams